UNIVERSITÀ DEGLI STUDI
DI MILANO

DIPARTIMENTO DI SCIENZE DELL'INFORMAZIONE
DOTTORATO DI RICERCA IN INFORMATICA DELLA
SCUOLA DI DOTTORATO IN INFORMATICA – XXII CICLO

TESI DI DOTTORATO DI RICERCA
INF/01

Investigations on cognitive computation and computational cognition

Isabella Cattinelli

Relatore: Prof. N. Alberto Borghese

Correlatore: Prof. Eraldo Paulesu

Il Direttore della Scuola di Dottorato in Informatica:

Prof. Ernesto Damiani

ANNO ACCADEMICO 2009–2010

Abstract

This Thesis describes our work at the boundary between Computer Science and Cognitive (Neuro)Science. In particular, (1) we have worked on methodological improvements to clustering-based meta-analysis of neuroimaging data, which is a technique that allows to collectively assess, in a quantitative way, activation peaks from several functional imaging studies, in order to extract the most robust results in the cognitive domain of interest. Hierarchical clustering is often used in this context, yet it is prone to the problem of non-uniqueness of the solution: a different permutation of the same input data might result in a different clustering result. In this Thesis, we propose a new version of hierarchical clustering that solves this problem. We also show the results of a meta-analysis, carried out using this algorithm, aimed at identifying specific cerebral circuits involved in single word reading. Moreover, (2) we describe preliminary work on a new connectionist model of single word reading, named the two-component model because it postulates a cascaded information flow from a more cognitive component that computes a distributed internal representation for the input word, to an articulatory component that translates this code into the corresponding sequence of phonemes. Output production is started when the internal code, which evolves in time, reaches a sufficient degree of clarity; this mechanism has been advanced as a possible explanation for behavioral effects consistently reported in the literature on reading, with a specific focus on the so called serial effects. This model is here discussed in its strength and weaknesses. Finally, (3) we have turned to consider how features that are typical of human cognition can inform the design of improved artificial agents; here, we have focused on modelling concepts inspired by emotion theory. A model of emotional interaction between artificial agents, based on probabilistic finite state automata, is presented: in this model, agents have personalities and attitudes that can change through the course of interaction (e.g. by reinforcement learning) to achieve autonomous adaptation to the interaction partner. Markov chain properties are then applied to derive reliable predictions of the outcome of an interaction. Taken together, these works show how the interplay between Cognitive Science and Computer Science can be fruitful, both for advancing our knowledge of the human brain and for designing more and more intelligent artificial systems.

To all the people who have always believed in me.

Acknowledgements

"Live all you can - it's a mistake not to.
It doesn't so much matter what you do in particular, so long as you have your life."
— Henry James, 1843–1916

Writing the acknowledgments for one's Thesis is always an emotional moment. You have reached the end of a journey, made of full dedication and passion, sacrifices and effort, and you finally get to see the fruits of your labor. Reaching the end of such a journey is in itself a truly rewarding moment; what is even more rewarding, however, is looking back to what brought you here, and realizing how many people supported you along the way.

The work in this Thesis would not exist if it were not for my supervisors and collaborators. In particular, I owe much to my advisor, prof. Borghese, who has guided and supported me with patience and genuine interest, and encouraged me to be autonomous and rely on my intuitions. I wish to thank my co-advisor, Prof. Paulesu, for introducing me to the field of neuroscience: studying the human brain has fascinated me to the point that it became the fulcrum of my Ph.D. work, and it is now at the top of my research interests. I am thankful to both my supervisors for having always shown an uttermost confidence in my abilities, which made me often desire to be able to look at myself through their eyes. I am also indebted to Prof. Plaut for warmly welcoming me in Pittsburgh during my semester as a visiting student, and for giving me the opportunity to work with him on a topic (connectionist cognitive modelling) that is extremely appealing to me. I am also grateful to my referees, who generously gave their time to read my Thesis and help improve it.

I wish to thank all those special people that I happened to meet at schools, conferences, during my travels, and that, sometimes unexpectedly, turned from interesting acquaintances into faithful friends with whom a genuine bond exists. I cannot name them all, but I am sure they know I am talking of them. I am thankful for having the opportunity of traveling, and also living abroad for a while: every journey has been an occasion for personal growth, and I have always come back with renewed enthusiasm for research.

After thanking the distant ones, I turn now to the near ones: the AIS-lab boys and the Bicocca girls, who endured my presence in the respective labs day after day, and the "second-floor elite" at DSI, who shared philosophical meals with me in the very last period of my Ph.D. All of them have always been incredibly supportive of me, much beyond of what I think I deserve. I shared with them my ups and downs, and I must thank them for bearing with me and my idiosyncrasies.

No acknowledgments could be complete and fair without mentioning my family. They are indeed my safety net and I know that, whatever happens and whatever my choices are, they will always be there for me. And finally, I want to thank myself, too: for, after all, I managed to get here, to the conclusion of this enterprise. And it is no small thing.

E quindi uscimmo a riveder le stelle.

Preface

A proper subtitle for this Thesis could have been: "The Eclecticism of Computer Science". In fact, computer science is a very flexible discipline whose tools can be successfully applied to tackle very diverse research questions, in many different scientific areas. Computational techniques are pervasively employed in virtually all branches of research for both automatized data analysis and modelling. More rarely, but not less importantly, research in computer science itself incorporates results from other areas to enhance the effectiveness of its techniques and tools. Cognitive science and neuroscience have been especially fruitful ground for this synergy with computer science.

On the one hand, neuroscience and psychology are increasingly taking advantage of computational tools to help advancing our knowledge of how the brain works, both on a small scale (single neurons, small neural populations) and on a larger one (cognition and behavior). Computational and statistical techniques are already widely used to rigorously interpret data collected by means of behavioral or neuroimaging studies (i.e. statistical testing, cluster analysis, image processing); also, the modelling power of computer science has proved useful to provide support (or refutation) for theories on cognitive and neural processing. Thus, computer science has been a valuable servant to the study of the brain and the mind. At the same time, the design of computational systems and algorithms has benefited from the advances in understanding the neural and cognitive basis of natural intelligence that were made possible by the work of neuroscientists and psychologists: since the richest, full-functioning intelligent system in nature is the human brain, it naturally fits the role of prime inspiration, and model, for building artificial systems that might hopefully approach, one day, similar levels of performance, flexibility, and robustness.

This Thesis embodies these principles of eclecticism, and mutual beneficial exchange between such disciplines. Three perspectives are taken, to each of which a separate part of the Thesis is dedicated:

1. computer science as the provider of fine-tuned tools for analyzing neuroscientific data;
2. computer science as an effective way to model, and therefore interpret, cognitive processes;
3. computer science as the science of building artificial intelligent systems inspired by features of human cognition.

Part I deals with the analysis of data from large collections of neuroimaging experiments, also called meta-analyses. Functional neuroimaging experiments have become, in the last 20 years, a major investigation technique to look into the functional-anatomical basis of cognition, with the aim of locating brain regions that are specifically involved in the cognitive process of interest. The meta-analytic approach allows researchers to combine information from several different studies, and extract from this extended dataset consistent and robust knowledge that can provide ground for novel conclusions on the topic at hand. Chapter 1 provides an introduction to basic neuroscientific notions, and presents the main ideas behind neuroimaging experiments and meta-analyses. One approach to the meta-analysis of functional neuroimaging data employs a clustering procedure to automatically group activation coordinates into clusters that can then be further analyzed to determine their functional role within the considered cognitive process. Adopting this approach, we worked in the direction of developing an improved meta-analytic methodology. In particular, we designed a variant of classical hierarchical clustering that successfully deals with the problem of non-uniqueness of the solution; this problem is not irrelevant, although usually neglected in applications, because it can potentially lead to different interpretations being given of the same data if these are permuted differently. Our algorithm ensures that the result of clustering a given dataset does not depend on the particular ordering of the input data. Our algorithm, together with a review of clustering techniques, and additional methodological contributions, is described in Chapter 2. Finally, Chapter 3 reports a complete meta-analysis on the neuroimaging of single word reading that we performed using our algorithm. Our meta-analysis offers a condensed picture of the circuits involved in the reading process, enforcing the notion of a widely distributed network where no clear-cut segregation of processes, based purely on the lexical status of the orthographic stimulus, can be observed.

Part II turns to the topic of computational modelling of cognitive processes as a valuable approach to gain insights into the mechanisms of cognition; in particular, we have focused on connectionist models that, unlike other classes of models, directly address the algorithmic (the *how*) questions, rather than restricting themselves exclusively to computational (the *what*) ones. Chapter 4 presents a broad introduction to modelling in cognitive science and to artificial neural networks. In particular, linking back to the theme touched in Part I, our interest is on modelling single word reading. After a review of existent models (see Chapter 5), we present in Chapter 6 our own modelling work, motivated by the question of whether serial effects in reading can be correctly reproduced in a inherently parallel system. A working framework based on the computation of an all-purpose code for word-related tasks is introduced, and the portion involving word production (i.e., mapping from semantics to phonology) was implemented as a connectionist network. This network hosts two conceptually different components: the cognitive part is responsible for computing the internal code representing the input word; the articulatory part implements the sequential production of the phonemic output. A threshold mechanism, based on the quality of the code being computed over time, regulates the flow of information between the two components and, thus, the timing of the

response. Several versions of this model, differing in implementational details of the threshold mechanism, and in training regimens, were implemented: although good accuracy on the training dataset was always achieved, concerns about the stability of learned representations and difficulties in reproducing frequency effects invited us to present a critical analysis of this effort, in an attempt to point out the possible reasons behind the observed behavior and to suggest implementational choices that might overcome these weaknesses. Future work will be needed to further assess the potentiality of this model in accounting for a significant range of effects in single word reading.

Part III illustrates how concepts and models from psychology (and neuroscience) can inform the design of intelligent artificial systems. We switch here from pure rational cognition (as represented by the process of single word reading that was considered in Part I and II) to emotion; this switch is only apparently abrupt, as emotions are not outside the domain of cognitive science (although, admittedly, emotional processing has not been a primary topic in this discipline). Emotions are deeply intertwined with "cold" cognition, and play an important role in decision-making, social interaction, learning, and motivation. The relevance of the influence of emotion on intelligence at large has been acknowledged also by computer scientists that, in recent years, have started to develop systems endowed with some form of emotional intelligence (either in recognizing, expressing, or modelling emotions). An overview on emotions in humans, and in machines, is given in Chapter 7. Finally, in Chapter 8 we report our own work in designing an emotional interaction model for artificial agents that can be used both in human-agent and agent-agent interaction scenarios. Our model, based on probabilistic finite state automata, computes the emotional state for an agent based on its personality and attitude toward the interaction partner. The latter changes over time to autonomously adapt to the particular partner: in particular, a reinforcement learning approach is proposed, which allows an agent to learn the most effective attitude to lead the other agent into specific (goal) emotional states. The resulting inter-agent interaction scenarios can be analyzed by resorting to Markov chain theory, so as to automatically derive the most probable emotional states experienced by the two agents as the interaction between them protracts. A possible application for this interaction model lies in supporting the design of virtual agents endowed with the ability to communicate with emotions, for instance in videogames; the possibility of producing different patterns of emotional behavior based on different initial personalities, and on the natural dynamics of the interaction itself (which guides the process of adaptation to the partner), would allow for natural and rich interactive experiences.

Although these three research lines might appear to be very diverse, they are nonetheless unified by one underlying theme: human intelligence. The works described in this Thesis share the same general objectives, in attempting to understand the basis of natural intelligence and to reproduce some of its aspects in artificial systems. The choice of investigating the faculty of reading, on the one hand, and affective behavior, on the other hand, is motivated by the observation that both are high-level functions that are highly distinctive of human intelligence. Reading is an exclusive human ability, and whereas affective interactions

do take place between other animals too, they rarely reach a similar degree of complexity and diversity of behaviors as those observed in the human society. For this reason, both reading and emotional interaction represent privileged observation points from which we might ultimately infer some of the main principles that inform human cognition.

In conclusion, this Thesis illustrates the potentiality of the synergy between computer science and cognitive (neuro)science from different points of view, advocating the relevance of interdisciplinary research as a powerful means to advance theoretical knowledge and foster the development of better technology. Without such fertile interplay, several significant advancements and important results would probably have never been achieved: let us think of artificial neural networks, whose design was inspired by the biological neural cell and which are nowadays powerful tools for many pattern recognition applications, or of the computational techniques that make it possible to perform neuroimaging experiments, turning bright voxels into meaningful pieces of information about the functioning principles of the brain. Our contribution goes in this direction, providing a collection of results (a novel clustering procedure, a meta-analysis of the neuroimaging of reading, an exploratory modelling work on reading, and a model for inter-agent emotional interaction) that were fostered by, and will hopefully promote, interdisciplinary research. Interdisciplinarity also translates into the opportunity of working with many diverse researchers, each giving their valuable contribution to collaborative work, such as the one described in this Thesis, and sharing their knowledge in a very fruitful exchange. It is therefore only right to acknowledge here every contributor to the materials presented next.

The algorithm dealing with the problem of non-uniqueness of the solution in hierarchical clustering described in Chapter 2 was presented in a paper that is currently being revised for journal resubmission (Cattinelli et al., Under Revision–b); in particular, the author of this Thesis wishes to thank Giorgio Valentini for performing the experiments on the bioinformatics dataset, and all co-authors (Valentini, Paulesu, Borghese) for the cooperative effort. The work on semantic clustering mentioned in the same chapter was carried out by Marco Radaelli for his Master's Thesis (Radaelli, 2010), under Prof. Borghese's supervision and in close collaboration with the author. The meta-analysis of functional neuroimaging data on single word reading (Chapter 3) is described in an article to be resubmitted in revised version (Cattinelli et al., Under Revision–a), reported here in its present form; the contribution of all co-authors (Gallucci, Borghese, Paulesu) was precious. The modelling work constituting Chapter 6 was conceived, and carried out for the most part, during the author's visiting period at Carnegie Mellon University in 2009; the author is indebted to Prof. David Plaut for his guidance and help on this work. Finally, Chapter 8 reproduces a published article (Cattinelli et al., 2008) written in collaboration with Prof. Massimiliano Goldwurm and Prof. Alberto Borghese. Any merit of this Thesis is therefore shared with all these people.

Contents

"Computer science is no more about computers than astronomy is about telescopes."
— Edsger W. Dijkstra, 1930–2002

Part I
Meta-analysis of neuroimaging data

Chapter 1

The quest for understanding the human mind, part 1: the functional imaging way

"As long as our brain is a mystery, the universe, the reflection of the structure of the brain will also be a mystery."

— Santiago Ramón y Cajal, 1852–1934

1.1 Understanding the human mind: the final frontier

As human kind progresses in the quest for expanding its knowledge of the universe, surprisingly little is known about the very source of such knowledge: the human mind. Arguably, the final frontier of human knowledge is represented by a full understanding of how our own mind works.

In this challenge, we are faced with a very complex system that can, and needs to, be studied at different levels. Marr (1982) sketched a general framework for describing and understanding any complex computational system, consisting of three levels of decreasing abstraction:

1. *Computational level* (**Computational theory**): it describes *what* the system does (in terms of the abstract properties of the computation that is carried out), and *why* it does so.

2. *Algorithmic level* (**Representation and algorithm**): it describes *how* the system performs its computation (the *what* of the previous level), and requires a choice of representations for the input and the output, and an algorithm that maps the former into the latter.

3. *Implementational level* (**Hardware implementation**): it describes the physical realization of the representations and the algorithm on the actual computational device.

Although a full understanding of the considered computational system can eventually be achieved only when all three levels have been mastered and linked together, and although a

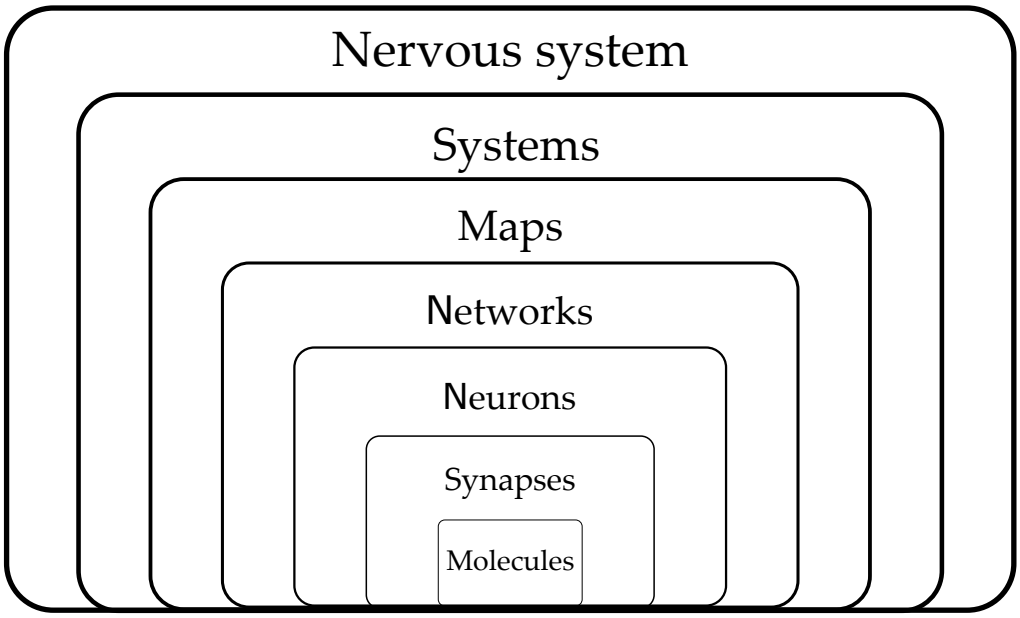


Figure 1.1: Different levels of organization in the brain anatomy (modified from Churchland and Sejnowski, 1994).

choice at one level can influence choices at the other levels, the levels are nonetheless quite independent, in that one can work at a single level mostly disregarding details of the remaining ones. As Marr says: *"any particular representation makes certain information explicit at the expense of information that is pushed into the background and may be quite hard to recover"* (Marr, 1982, p. 21). On the one hand, focusing on one level of analysis simplifies the problem and helps leave out less relevant details. On the other hand, this implies that the choice of the level(s) one intends to work at is crucial, as specific phenomena might be better understood by targeting a specific level and, once the choice is made, details regarding the other levels are lost.

As any other complex computational system, the human brain is suitable to be analyzed at multiple levels. This analysis can be performed in Marr's sense: different levels correspond to different sets of questions (e.g. what is the purpose of the visual system? vs. how is orientation sensitivity implemented in V1 neurons?). A level-based analysis can also be carried out by proceeding through levels of organization in the brain (Churchland and Sejnowski, 1994, chap.2). The anatomical organization of the brain is in fact hierarchical, with phenomena occurring at different scales, from molecules up to the entire nervous system (Figure 1.1). Although one may be tempted to pair the lowest anatomical scales with the implementational level, and higher scales with the computational level, such pairing would only be a loose one. In fact, each anatomical level can be looked at from different standpoints, e.g. both from the computational and the implementational level, depending on the nature of the questions we are trying to address at that particular scale (Churchland and Sejnowski, 1994, chap.2).

In an even more simplified framework, we can distinguish between approaches to the

study of the mind that focus on its anatomical substrate, the brain; and approaches that, on the other hand, focus on its *output*, that is, behavior and mental states supporting it. Roughly speaking, we have therefore two major levels of analysis, with the "brain" level being, arguably, the lowest one in terms of abstraction degree.

In this very schematic account, neuroscience operates mainly at the lowest level. The functioning of single neurons, or of networks of neurons, is investigated through a range of techniques, from patch clamp to electrophysiological recordings, from lesion studies to functional neuroimaging, etc. Some of them are *in vivo* techniques, studying whole, living systems (as in neuroimaging), others are *in vitro* (e.g. patch clamp). The growth of computational neuroscience (Dayan and Abbott, 2001) has also provided a way to study neuronal functioning *in silico*, that is through simulations of biologically realistic models of nervous cells.

More abstract levels of analysis are carried out by (cognitive) psychology. Here, the brain is studied as a sort of black box to which stimuli are provided, and whose outputs in response to such inputs are observed. The observed association between input-output pairs, and relative measures of performance, are used as a basis for inferring the characteristics of the computation performed by the black box. For instance, reaction times recorded when subjects are asked to perform a given task can be related to the complexity of the task itself: the assumption here is that more complex tasks require longer reaction times. Neuropsychological studies offer also the opportunity to compare different black boxes: the healthy brain vs. the brain suffering a deficit (caused by a lesion or by a neurological disease). The comparison of the observed behaviors of these two black boxes can help shed some light on the nature of the computation carried out by the brain for the task at hand. Therefore, psychology mostly addresses the issue of unveiling the computational theory of the brain. Cognitive theories often translate to simplified box-and-arrow models that represent the set of involved processes and their interactions. Besides the experimental approach, additional insights can be provided by computational psychology (see Chapter 4): here, actual implementations (*computational models*) of cognitive theories are developed, and tested against experimental data to determine the degree of fitness of the model, and of the theoretical assumptions it rests on. Computational models can be especially precious in that they help investigate in detail lower levels of abstraction, namely the algorithmic level and, to some extent, the implementational one. In fact, in a computational model both input-output representations and algorithms need to be explicitly declared (differently from classical box-and-arrow models), and the appropriateness of these choices in producing the desired computation can therefore be directly assessed.

Thus, a large range of instruments and approaches exist that allow us to work at different levels of abstraction in our quest for understanding the human mind. Some advances in this direction have already been made; much more remains to be understood. In the remainder of this chapter, we shall provide a brief overview of functional neuroimaging, one of the most used techniques for studying the link between cognition and brain activity, that is between the *what* and the low-level *how*[1] of the amazing computational device that is our brain.

[1]It is fair to say, however, that, especially in classical, univariate approaches to the analysis of neuroimaging

1.2 The human brain

1.2.1 The fundamental computational unit of the brain: the neuron

The basic computational units in the brain are the *neurons* which, together with the glial cells, constitute the nervous tissue (Bear et al., 2001). The study of neural cells originates back to the nineteenth century, when some important technical advancements (the invention of the microtome, and especially the introduction of Nissl's and Golgi's stains) made it possible to isolate single neurons and recognize their constituent parts. Key figures in those early days of neuroscience were Camillo Golgi (1843–1926) and Santiago Ramón y Cajal (1852–1934) who were jointly awarded, for their efforts in advancing knowledge on the nervous system, the Nobel Prize in Physiology in 1906. They had, however, different views on the basic organization of the nervous system: Golgi believed it to be a syncytium, that is a unique, large cell made up of different nuclei, whereas Ramón y Cajal supported the *neuron theory* according to which different, separated cells can be distinguished in the brain, as in any other biological tissue. In fact, the gap existing between single neurons ($\approx$ 20 nm) could not be observed with the microscopes available at that time, and only the introduction of the electron microscope (having a resolution of 0.1 nm) in the 1950s provided the final evidence for the neuron theory.

Observing a neuron at the microscope reveals that it is composed of several parts, each with a specific function. The body, or *soma*, contains the cell's nucleus and is the site of the metabolic processes sustaining the life of the cell; in particular, the mitochondria in the soma transform oxygen and glucose into molecules of adenosine triphosphate (ATP) that provide energy to the neuron, in particular for supporting the processes involved in the generation of electric signals. From the soma several outgrowths depart, that are collectively called *neurites*. Among these, we can distinguish between the *axon*, the output projection of the neuron, and the *dendrites*, its input terminals. The role of a neuron consists in receiving, combining, and transmitting messages within the neural network; such messages are in the form of electric pulses, called *action potentials*, or *spikes*. Thus, action potentials are received from other neurons (named pre-synaptic) via the contact points (*synapses*) established on the dendrites of the post-synaptic neuron. If the incoming integrated signal is strong enough, it elicits the generation of an action potential by the post-synaptic neuron; this signal is then propagated along the axon of the neuron to its terminals. The axon of a neuron can be very long (up to 1 m in humans) and its transmission speed is remarkable (it can vary from 0.5 m/s to 120 m/s); speed and accuracy in signal transmission are achieved by the combination of an insulation mechanism (axons in the nervous system are covered in a *myelin sheath*, which is provided by glial cells) and the existence of nodes for signal regeneration (*nodes of Ranvier*).

data, we should more precisely speak of *where* – in fact, the location of the most active brain regions for a given experimental task is what is usually investigated in neuroimaging studies. Notice, on the other hand, that more recent techniques are actually shifting the focus on the *how*, by studying the relationship between patterns of activation over a restricted neural population, and the mental state that elicited them. For reviews on multi-voxel pattern analysis methods, the reader can refer to (Haynes and Rees, 2006; Norman et al., 2006; Pereira et al., 2009; Mur et al., 2009).

The neuron is enclosed by a phospholipidic membrane that separates the intracellular fluid (called *cytosol*) from the extracellular one. Both contain ions, but with different concentrations: in the rest condition, the cytosol is characterized by high concentration of K^+, whereas the extracellular liquid is richer in Na^+, Ca^{++}, and Cl^-. Because of this *concentration gradient* existing at its sides, the neuronal membrane has a negative resting potential ($\approx -65\ mV$); in the rest condition, this potential is maintained by the equilibrium between the diffusion force, that pushes ions to flow through the membrane to balance the concentration gradient, and the electric force, whereby a negatively polarized membrane attracts back escaped positive ions. Ions flow inside or outside the neuronal membrane via (Na^+ and K^+) ion-selective channels.

The arrival of a signal at the pre-synaptic site causes the opening of additional Na^+ channels; the resulting inward flow of ions determines a change in the membrane potential (called *depolarization*, because the potential goes from being negative to being positive, or less negative). If this depolarization (called excitatory post-synaptic potential – EPSP) gets over a threshold ($\approx -40\ mV$), then an action potential is generated. This happens because the intense depolarization activates special voltage-dependent channels that exist in the initial section of the axon, the axon hillock (and at every node of Ranvier): first, the opening of sodium ion channels further increases the depolarization of the membrane (which reaches about $40\ mV$); then, their inactivation paired with the opening of potassium ion channels leads to rapid hyperpolarization ($\approx -80\ mV$) and, finally, restoration of the resting potential. It is this rapid fluctuation in the membrane potential that constitutes an action potential. The post-synaptic potential is the result of integration, in both time and space[2], of incoming signals: the more intense the incoming stimulation, the higher the frequency of generated action potentials[3]. If the post-synaptic potential is not sufficient for triggering a spike, it just decays. An action potential is, therefore, an all-or-none phenomenon, as opposed to the graded nature of the post-synaptic potentials, which can assume variable amplitudes. Post-synaptic potentials are also subject to degradation, in that they tend to dissipate as they move from the dendrites to the axon hillock.

The action potential is transmitted through the myelinated axon through the mechanism of saltatory conduction, whereby the flux of ions entered through the channels just passively diffuses along the myelinated section of the axon, and the action potential is re-generated at each node of Ranvier by the opening of voltage-dependent channels. The depolarization of the axon terminal, in turns, determines the opening of Ca^{++} voltage-dependent channels. The intake of calcium liberates from its vesicles, out in the synaptic cleft, a neurotransmitter, that is a chemical substance that diffuses toward the post-synaptic neuron. There, the neurotransmitter triggers the opening of ion channels in the dendrite membrane, and a post-synaptic potential is started. After an action potential has been generated, the ion concentration levels

[2]Time integration occurs when sustained stimulation is delivered at the same synapse; space integration refers to signals being delivered at different synapses of the post-synaptic neuron.

[3]The neuronal membrane is characterized by refractory periods which bound the spike frequency to a maximum of $1\ mH$.

of the cell at rest are actively restored by the ionic pumps, which push ions against their concentration gradient. Such process is very expensive for the neuronal cell, and in fact most of the ATP produced by mitochondria is employed for the operation of the pumps (especially the sodium-potassium pump). By this complex, and fascinating, electrochemical mechanism information is effectively propagated through the nervous system.

Although we have only mentioned EPSPs, IPSPs (inhibitory post-synaptic potentials) do exist, too. Inhibition corresponds to an hyperpolarization of the post-synaptic neuron, so that the generation of an action potential is hindered. This is due to an incoming flux of Cl^- ions through transmitter-dependent channels that are usually activated by the neurotransmitter GABA (gamma-aminobutyric acid). Other neurotransmitters, such as the glutamate, have generally an excitatory effect in that they trigger the opening of sodium ion channels. Yet other neurotransmitters (e.g. acetylcholine – ACh) can have either an excitatory or inhibitory effect, depending on the kind of receptor they interact with.

Although neurons can be divided in classes according to one of multiple criteria (such as number of neurites, length of the axon, released neurotransmitter), they all share the same basic functioning mechanism; the meaning of the computation carried out by a given neuron does not so much depend on the physical characteristics of the cell itself, but rather on the kind of connections that involve it (e.g. neurons innervating the muscular tissue transmit motor commands). Moreover, all neurons generate action potentials that are virtually identical in duration (about 2 *ms*) and amplitude; information is therefore carried not by a single action potential, but by a *train* of action potentials, whose frequency and structure are the truly informative features.

1.2.2 A brief look into the anatomy of the human brain

About 100 billion neurons are estimated to form the human nervous system. As the reader will see, looking into its organization resembles progressively zooming in on a picture to reveal, at each zoom step, a greater degree of detail about the involved structures.

The human nervous system can be, in the first place, divided into the central nervous system (CNS) and the peripheral nervous system (PNS). The brain and the spinal cord constitute the CNS, whereas the PNS includes the axons of motor neurons and the dorsal root ganglia that collect sensory information (somatic nervous system), and neurons that provide innervation to the internal organs (autonomic nervous system). The bodies of neuronal cells constitute the *gray matter*; axon fibers collectively form the *white matter*, the color being provided by the myelin sheath.

The brain can itself be divided into forebrain, brainstem, and cerebellum (Figure 1.2). The *cerebellum*, located posteriorly, is mainly a site for motor control; the *brainstem* is involved in several vital functions such as breathing, and represents a relay of information to other stations in the nervous system. The forebrain, which lies dorsally to the other two parts of the brain, is organized into two mostly symmetrical hemispheres, the left and the right *cerebral hemispheres*. Higher cognitive processes are carried out in the forebrain, and in particular in

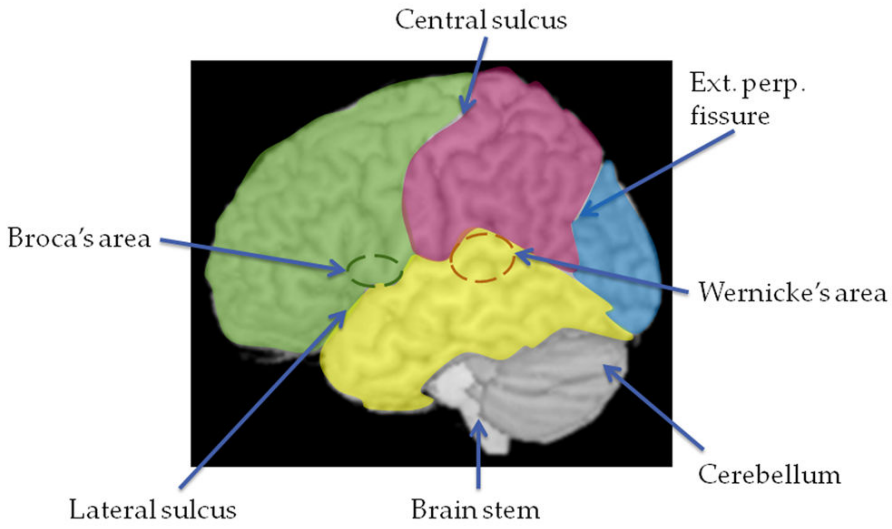


Figure 1.2: A lateral view of the human brain (left hemisphere). The forebrain, the cerebellum, and the brainstem can be seen. The figure highlights the main fissures and the subdivision in lobes: in green, the frontal lobe; in yellow, the temporal lobe; in purple, the parietal lobe; and in blue, the occipital lobe. Broca's and Wernicke's areas are also shown.

the *cerebral cortex*: this is a thin layer of gray matter that extends to the surface of the forebrain. Subcortical gray matter includes, among the others, the thalamus, the hypothalamus, the amygdala, and the basal ganglia (caudate nucleus, globus pallidus, and putamen). The white matter in the forebrain comprehends commissural fibers (the main ones are the corpus callosum and the anterior commissure) spanning the two hemispheres, the association fibers (the superior and inferior longitudinal fasciculi, the uncinate fasciculus, and numerous other shorter fibers) that link regions within the same hemisphere, and the projection fibers (internal, external, and extreme capsule) that collect axons connecting the cortex to other structures. The brain also hosts the ventricular system, which produces the cerebrospinal fluid (CSF) flowing beneath the subarachnoid space between the meninges; together, the meninges and the CSF serve as a protection mechanism for the CNS.

The cerebral cortex can be further partitioned into *lobes* (Fig. 1.2), and lobes into convolutions, or *gyri*. In humans, and primates, the cortex has the distinctive feature of being folded onto itself several times, so that the geometry of the brain surface is characterized by protrusions (gyri) and clefts (*sulci*); in this way, the cerebral cortex could considerably develop its extension through the course of evolution without requiring a proportional increase in the size of the skull. The subdivision into lobes is based on the landmarks provided by the main sulci, also called *fissures*. Looking at the lateral surface of the cortex, an approximately vertical fissure can be distinguished, named *central sulcus*, or *fissure of Rolando*: the *frontal lobe* is located anteriorly to this sulcus, which divides it from the *parietal lobe*, lying posteriorly. A mostly horizontal sulcus, the *lateral sulcus* or *fissure of Sylvius*, separates these lobes from the

more ventral *temporal lobe*. Both the parietal and the temporal lobe are posteriorly delimited by the *external perpendicular fissure* (and its ideal ventral prolonging); the *occipital lobe* is located posteriorly to this fissure. In addition to these four lobes that can be distinguished in a lateral view of the brain, a fifth lobe, the *insula*, can be found more medially, within the lateral sulcus.

Minor sulci subdivide each lobe into a number of gyri: for instance, on the lateral surface of the frontal lobe, four gyri can be counted, three of which with antero-posterior course (the superior, middle, and inferior frontal gyrus), and the more posterior precentral gyrus that runs parallel to the central sulcus. Further subdivisions are possible, as well: the inferior frontal gyrus is composed, in fact, of three parts (pars orbitalis, pars triangularis, and pars opercularis), the latter of which corresponds, in the left hemisphere, to Broca's area, a region involved in speech output, as first suggested in (Broca, 1861) on the basis of the non-fluent pattern of aphasia[4] observed in a patient suffering from a brain lesion in this region. Another brain region crucial for language abilities is Wernicke's area, which is located in the posterior part of the left superior temporal gyrus; its lesion was shown to be associated with fluent aphasia (Wernicke, 1874), from which the role of this area in decoding auditory input could be inferred. Wernicke's area and Broca's area[5] (Fig. 1.2) are connected by the superior longitudinal fasciculus; interruption of these fibers can result in selective deficits in the repetition of heard words (conduction aphasia).

For reasons of space, we have given here only a small sample of the anatomical regions that have been defined in the brain; many more gyri, on the lateral, ventral, and medial surfaces of the brain complete the picture of the cerebral cortex. Anatomical regions in the brain can therefore be identified on the basis of the gyral subdivision sketched above[6]. Alternatively (or additionally), a cytoarchitectonic classification can be employed. Korbinian Brodmann (Brodmann, 1909) isolated about 50 areas in the mammalian brain, that differ in their laminar organization. Brodmann's areas are identified with numbers: for instance, area 17 corresponds to the more posterior part of the occipital lobe and coincides with the primary visual area ($V1$), the first cortical station of the visual pathway. Broca's area corresponds to Brodmann's area 44 (and 45). Figure 1.3 reports both the AAL (Automated Anatomical Labelling – Tzourio-Mazoyer et al., 2002) and Brodmann templates, as employed in the visualization software MRICro (Rorden and Brett, 2000). Each color represents a different area: for instance, in Fig. 1.3(a) the dark green area in the frontal region corresponds to the pars triangularis of the inferior frontal gyrus, and approximately the same region is labelled as

[4]The term aphasia refers to an acquired disorder of the language faculty. Non-fluent aphasias are characterized by a deficit in articulation, whereas auditory comprehension is mostly spared; the opposite pattern is observed in patients affected by forms of fluent aphasia.

[5]The example provided by non-fluent and fluent types of aphasia and associated sites of lesion allows us to highlight the importance of the concept of *double dissociation* is neuropsychology: a double dissociation is observed when a lesion in area a' impairs function x', but not function x'', and a lesion in area a'' impairs x'' but not x'. The presence of a double dissociation is a strong indication that a' is actually the anatomical substrate of function x', and a'' of function x''.

[6]A classification based on gyri is suitable for cortical regions only; for subcortical structures, appropriate anatomical labelling is available as well.

(a) (b)

Figure 1.3: An axial section of the human brain is shown, with each region having a color coding for its anatomical (a), or cytoarchitectonic (b) label. Both images are available in the visualization software MRICro (Rorden and Brett, 2000).

area 45/47 under the Brodmann template (Fig. 1.3(b)).

A point in the cerebral volume can thus be described in terms of the anatomical, or cytoarchitectonic, region to which it belongs. The employment of a *stereotactic space*, however, permits a more precise localization of a specific point in the brain volume, and is especially useful in neuroimaging experiments (see next section). The Talairach and Tournoux (1988) space is a system of three-dimensional coordinates whose origin is determined by the intersection of the mid-sagittal plane, that is the ideal plane that separates the two hemispheres, with the straight line joining the middle points of the anterior and of the posterior commissure; the origin is located at the anterior commissure (AC). Each location in the brain is thus identified by an x coordinate that expresses the distance of the point from the AC in the left-right direction (negative values corresponding to the left hemisphere); by an y coordinate, which gives the distance from the AC in the antero-posterior direction (negative values corresponding to posterior loci); and by a z coordinate that designates the distance from the AC on the dorso-ventral axis (negative values corresponding to more ventral positions). The use of a shared reference space makes data produced by different laboratories (and derived from different brains!) more easily comparable. Stereotactic coordinates refer to a *prototypical* brain, an *atlas* or template, to which individual brains must be mapped in order to make results comparable. The Talairach and Tournoux (1988) atlas has been widely used in the past, but in more recent times the MNI (Montreal Neurological Institute) template has been employed more frequently as it is more representative of the "average" human brain.

1.3 Neuroimaging

1.3.1 Neuroimaging techniques

The advent of imaging techniques applied to the brain represented a turning point in neuroscience, because for the first time it became possible to look at the brain while in full functioning – the best that could be done before, was to study the brain of dead subjects, by analyzing its cytoarchitectonic structure, or, in former patients, observing the location of the lesions that gave rise to their deficit. Imaging techniques, on the other hand, made it possible to take pictures of the anatomy of a living brain (*structural imaging*), and even of the changes in its activity when performing experimental tasks (*functional imaging*).

Techniques for looking at the interior of the human body started to be developed at the end of the nineteenth century, when the discovery of X-rays by Röntgen provided a way to obtain images of the bones (and some kinds of soft tissue of the human body, such as the lungs); the very first X-ray "medical" image was that portraying the hand of Röntgen's wife. X-rays are a kind of electromagnetic radiation that is absorbed by some biological tissues, especially by bones; by placing a radiographic film besides the body being irradiated, the non-absorbed X-rays impress the film; the non-impressed portion of the film carries therefore an image of the bones.

An X-ray image, however, offers only a 2-dimensional view of an object; in the case of the brain, it is much more interesting to explore its 3D structure, and to be able to observe different sections of it. This was made possible with the introduction of *computed tomography* (CT), independently developed by Godfrey Hounsfield and Allan Cormack in the earlier 1970s. CT is based on the idea of taking images of the same object from different angles, and using them to create a 3D image of the scanned object. The X-rays tube (and the sensor, located on the opposite side) rotates around the object, resulting in multiple acquisitions of radiographic data; these are then processed by a computer via tomographic reconstruction, whereby sections of the imaged object are returned. A 3D reconstruction can then be obtained by combining these slices into a complete volume.

One drawback of CT in medical imaging is that it requires the body to be irradiated with X-rays, which can represent a health hazard. A more recently introduced imaging technique, on the contrary, does not employ X-rays and, for this reason and because it offers clearer results when imaging soft tissue, is being increasingly adopted in diagnostic procedures in place of CT scans. *Magnetic resonance imaging* (MRI) exploits the magnetic properties of hydrogen atoms to obtain structural images. An electromagnetic field is induced inside the MRI scanner, which has the effect to invert the spin of the hydrogen protons in the body water; when the electromagnetic field is removed, protons go back to their original state, releasing photons that can be detected by the scanner. Since different tissues (e.g. white and gray matter) have different magnetic properties, the rate of decay for protons in these tissues is also different; such difference is captured by the MR image. As in CT imaging, a complete brain volume can be obtained.

Both CT and MRI are useful techniques for obtaining structural images of the brain: they can be used, for instance, to localize lesions, or to investigate the density of gray matter in different populations of subjects. However, these images do not tell anything about the operations of the brain itself. For this, one must resort to functional imaging. Among the functional imaging techniques, the most popular ones are the Positron Emission Tomography (PET) and the functional MRI (fMRI) (Raichle, 1994, 2003). Both measure changes in the local metabolism of the brain, based on the assumption that an increase in energy consumption corresponds to intense neural activity occurring at that locus. An active neuron requires more energy, and this determines an increase in regional cerebral blood flow (rCBF) so that more oxygen and glucose can be brought to active areas. PET measures glucose levels in the rCBF; recall that ATP, the molecule providing energy to the neuronal cell, is produced by consuming glucose. An increased neural activity is thus accompanied by an increase in glucose supply. Glucose molecules are tracked by injection in the blood circulation of a radioactive tracer, whose presence can thus be detected by the scanner sensors. fMRI, on the other hand, registers the ratio of oxygenated hemoglobin (oxyhemoglobin) to deoxyhemoglobin in the blood – the so called Blood Oxygenation Level Dependent (BOLD) signal (Logothetis and Pfeuffer, 2004). This is possible because oxyhemoglobin and deoxyhemoglobin have different magnetic properties. Regions with a stronger BOLD signal are characterized by a higher concentration of oxygen in the blood, and are therefore more likely to be active. fMRI has several advantages over PET: first, it does not require any contrast medium to be used, and is therefore completely non-invasive; second, it has better spatial resolution (2-3 mm^3 vs. 5-10 mm^3); third, it usually requires less computational time for image acquisition. Both techniques are currently being used in neuroimaging laboratories spread across the world, although fMRI is now the leading approach.

1.3.2 Neuroimaging experiments

Functional neuroimaging techniques are employed in experiments aimed at identifying the anatomical substrate of cognitive processes of interest. To this end, experimental tasks are carefully designed, and assigned to groups of subjects that have to execute them while lying inside the scanner. The scanner will return a functional image representing the degree of activation (in fMRI, this is the BOLD signal) of each voxel (that is, a volume element) in the brain, recorded while the subject was engaged in the task. Such image alone, however, would not be very informative; it would not be possible to distinguish between areas that are specifically involved in the processing of the given task from those that would be active anyway. For this reason, many neuroimaging experiments adopt the *subtraction paradigm*: by subtracting the image obtained when the subject was performing task T_1 from the image corresponding to task T_2, a difference image is obtained that shows the brain areas that are *more involved* in task T_2 than in task T_1. Usually, task T_1 is a control task, or rest condition, such as passively looking at a screen; in this case, the difference image represents all those areas that are involved in the execution of task T_2 after canceling out "background" activity. The

result of a functional neuroimaging experiment is thus a set of areas of activation obtained in response to the administered experimental manipulations.

Designing a functional neuroimaging experiment requires many decisions to be taken. One or more group of subjects, having particular characteristics, must be recruited; for instance, we might want to compare brain activations under a given experimental tasks in male vs. female subjects, in adults vs. children, in neuropsychological patients vs. control subjects, or, on the other hand, focus on a group of average normal subjects to get a more general picture of normal cognition. Experimental tasks, or conditions (that is, the same task is used but on different classes of stimuli, such as looking at pictures of animals vs. pictures of inanimated objects), and control tasks must be carefully chosen, so that the subtraction paradigm can effectively isolate the cognitive subprocesses of interest: for instance, to identify the anatomical substrate of color processing, subjects can be shown colored blocks, alternated with grayscale blocks, and the difference image between the two conditions can be used to infer areas specifically involved in color processing; to highlight regions devoted to motion processing, moving objects can be contrasted to the static version of the same objects (Zeki et al., 1991). Stimuli used in the experimental tasks must be chosen carefully, so that all possible confounding variables are balanced: for instance, in the color experiment just described, the luminance of grayscale blocks was controlled so that it was matched to that of color stimuli; in this way, observed differences in brain activation when contrasting the two conditions cannot be ascribed to differences in luminance. Moreover, experimental conditions can be presented to subjects in different modalities. A block design can be used: the experiment is organized in blocks, where each block contains stimuli for one experimental condition (usually differently randomized within the block for each subject), and a sequence of such blocks makes up the experiment. In this case, the functional image for an experimental condition is an average of the brain activation observed during the processing of all the relevant blocks. In an event-related design, on the other hand, stimuli from all conditions are randomly intermixed, and their order of presentation must be recorded so that the activations corresponding to one condition can then be retrieved.

Difference images obtained in a neuroimaging experiment, however, are not, per se, reliable sources of information about neural circuits. Observed activations might result from noise, for instance. Quantitative methods are required to ascertain which sets of results are statistically significant and which others must be discarded as spurious. For this reason, the processing of raw data coming from the scanner is performed by dedicated software packages, that couple image preprocessing operations with statistical analyses to produce the final set of results. These usually take the form of sets of activation coordinates in the stereotactic space, corresponding to statistical comparisons of interest as selected by the experimenter.

A popular approach for analyzing neuroimaging data is the Statistical Parametric Mapping method (SPM), which is implemented in the homonym software[7] (Friston et al., 1990, 1995, 2007). The main steps of a typical SPM analysis are:

[7] Website: *http://www.fil.ion.ucl.ac.uk/spm/*.

1. image realignment;
2. spatial normalization;
3. smoothing;
4. fitting to the General Linear Model (GLM);
5. statistical testing and construction of a statistical parametric map;
6. statistical thresholding.

The first three steps do not actually involve statistical manipulations, but are meant to prepare the data (that is, the images returned by the scanner) for further processing. Realignment is carried out to correct for possible movements of the subject while in the scanner; in this way, all scans for the same subject are comparable. However, in a typical experiment we have a number of subjects, and we want to be able to compare the data for all of them. Thus, we need to perform image normalization, so that acquisitions related to single subjects are all brought into the same reference space. A smoothing step follows, with the purpose of increasing the signal-to-noise ratio for the subsequent statistical analysis. These preprocessing steps are carried out by using traditional imaging processing algorithms.

Statistical modelling follows. Volumes returned by the scanner are fitted to a GLM:

$$Y = X\beta + \varepsilon \tag{1.1}$$

where Y is the matrix of all voxels in all volumes, X is the *design matrix*, β are the parameters being estimated, and ε is the error term. The critical term here is the design matrix, which describes the relevant variables in the experiment, as well as confounding variables. The columns of the design matrix are called regressors. Typically, we will have a regressor for each experimental condition: in other words, we will have at least as many columns in the design matrix as there are conditions in our experiment, and each column will indicate whether a given volume was obtained under the corresponding experimental condition. Confounding covariates like global activity are usually modelled, too. It is also worth mentioning that SPM also allows the shape of the hemodynamic response to be taken into account when analyzing fMRI data. In fact, the increase in the BOLD signal does not occur synchronously with stimulus presentation: there is a delay of about 5 s before a peak in the signal occurs, followed by a persistent undershoot. Therefore, to model the activation data more precisely, SPM allows the time series for each condition (modelled as a collection of delta functions, or as a box-car function, depending on the experimental design) to be convolved with the hemodynamic response function before entering the design matrix as a regressor.

The β parameters of the GLM, one for each voxel, are estimated in a least-square fashion. At this point, the estimated $\widehat{\beta}$ values are used to perform statistical inferences to extract the effects of interest. This is done by defining statistical comparisons, called *contrasts*, between conditions. For instance, one may want to compare brain activity in response to seeing colored

blocks (condition c_1) vs. seeing grayscale blocks (condition c_2). This would correspond to the contrast vector $c = [1, -1, 0, \dots]$; recall we are subtracting activations related to condition c_2 – modelled by the second column in our hypothetical design matrix) from those related to condition c_1 – and other potential conditions do not enter our comparison. A t-test will be performed on the contrast $c^T\widehat{\beta}$ for each voxel; the resulting t-scores can then be reported on a statistical parametric map. Similarly, an F-test can be used whenever we are interested in finding all areas that are responsive to at least one of a number of conditions.

The statistical parameter map for the contrast of interest is now ready to be submitted to a process of statistical inference. Basically, what is done at this point is a thresholding step whereby only those voxels having a large enough t-score are returned as statistically significant with respect to the effect under study. For instance, we may want to obtain all voxels that are activated at a P-value not greater than 0.01; in other words, assuming that the null hypothesis holds – that is, no significant difference exists in the activations associated with the considered experimental conditions – we are requiring that there is, at most, a probability of 1% that we produce a false positive. This statistical approach is univariate in that a separate test is performed for each voxel. As many tests must thus be computed, the significance level for single voxels might be distorted as a result: this is known as the multiple comparisons problem. When testing discrete data, the Bonferroni's correction is usually employed, that consists in dividing the desired P-value by the number of total comparisons, and use the resulting value as the statistical threshold for significance. In the case of an SPM, correlation between contiguous voxels makes Bonferroni's approach too conservative; in its place, Gaussian Random Field theory can be used, which can be seen as the continuous analog of Bonferroni's correction. At the end of the SPM pipeline, we are left with an image (sometimes called t-map) where voxels that did not survive statistical testing are turned off; it is therefore possible to identify which voxels were significantly more activated in condition c_1 than in condition c_2, along with the magnitude of the effect of interest.

At the end of the first-level analysis described thus far, we have a summary image for each contrast of interest and for each subject. A second level analysis is then usually run, that consists in feeding these first-level images as data (values in Y) so that group inferences can be made. The final result is a set of coordinates – also referred to as *activation peaks* – on the brain volume that are significantly more active in a condition with respect to another, consistently over all tested subjects; each activation peak is also associated with its corresponding signal magnitude. Depending on the cognitive process that is assumed to lie in the difference of the two contrasted conditions, it may be concluded that those peaks identify the areas representing the anatomical substrate for that cognitive process.

1.4 Meta-analyses of functional neuroimaging studies

Each functional neuroimaging experiment brings about new insights into the anatomy of cognition, and permits to refine our knowledge about the dynamics of the human brain. How-

ever, there are obvious limits to the amount of knowledge that can be extracted from a single experiment: to clarify additional research questions, new ad-hoc experiments need to be designed and run.

Nonetheless, the utility of a neuroimaging experiment is not exhausted with the publication of the related paper. It is in fact possible to re-use the published data available in the literature to perform a *meta-analysis*. Like the word itself suggest, in a meta-analysis previously analyzed data are reassessed: this is done by collecting activation data from several related neuroimaging experiments (for instance, a collection of neuroimaging studies on single word reading), and looking into this extended, collective dataset to find the most consistent results, that is the ones that have been consistently replicated across studies, as determined by a sound, quantitative evaluation. The meta-analysis approach has thus confirmatory power with respect to those results that are more likely to be reliable, at the same time removing less robust results from the overall picture. Moreover, by working on a dataset of a larger size than what is allowed by a single study, the meta-analysis enjoys a greater statistical power, which may help extract novel knowledge about the cognitive aspect of interest that could not be unveiled by single, less powerful studies.

Meta-analyses are increasingly becoming popular in the neuroimaging field, thanks to the impressive amount of available activation data published in the literature in the past 20 years. This makes classical qualitative reviews of the literature less viable, and rather calls for more quantitative approaches. These, moreover, have the desirable quality to be more objective, in that the personal biases of the investigator are less likely to influence the conclusions of the review. However, the role of the researcher is still of primary importance, especially in the selection of candidate experiments to be included in the meta-analysis, in the definition of the variables and effects to be investigated, and in the critical interpretation of the results.

In order to provide an unitary view, we can list five main steps that constitute any meta-analytic process:

1. identification of the cognitive-physiological process(es) to be investigated;
2. establishment of inclusion criteria for studies, contrasts, and (possibly) single activation peaks;
3. data collection;
4. aggregation of the data to obtain a summary of the results;
5. extraction of significant results, and discussion on their interpretation.

Thus, an investigator must first choose the research domain he is interested in – for instance, motor imagery. Potentially, every scientific article that describes a functional neuroimaging study where an experimental task involving motor imagery is employed, could enter the meta-analysis. However, the investigator might want to exclude some studies to enhance the homogeneity of the dataset: for instance, he might want to use only activation coordinates coming from *direct contrasts* (e.g. imagining to move your hand $>$ imagining to move

your foot) rather than including also *simple contrasts* (e.g. imagining to perform a movement $>$ rest), the latter being deemed less specific. Once all inclusion criteria have been established, relevant articles must be scrutinized to extract activation data that match these criteria; a database will result, generally reporting for each peak all the relevant information about the experiment that generated it. At this point, peaks in the dataset are combined: this is where different meta-analytic approaches actually make different choices. In fact, several meta-analytic techniques, differing in their implementation of points 4. and 5. in the list above, have been proposed in the neuroimaging literature (Wager et al., 2007) – for an idea of the many techniques employed in this field, see Table 2 in (Wager et al., 2009). For purposes of illustration, we can here distinguish two main classes: the kernel-based approaches and the clustering-based ones.

In the first family, we can find ALE (Activation Likelihood Estimate: Turkeltaub, 2002); AGES (Aggregated Gaussian-Estimated Sources: Chein et al., 2002); KDA and MKDA ((Multilevel) Kernel Density Analysis: Wager et al., 2003, 2008). These methods are based on the idea of convolving each activation peak of the meta-analysis dataset with a gaussian (ALE) or spherical (KDA) kernel, to build blobs of spread activation containing one or more peaks. The obtained map is then compared to a large number of randomly generated maps (in a typical Monte Carlo fashion), to check for the significance of these blobs against such null distribution: a statistical test is performed to reject the null hypothesis that the map generated from the collected peaks could be the result of a random generation process, rather than representing a set of consistent (replicating) results. The surviving blobs are therefore the regions of greater consistency across the studies that entered the meta-analysis. The above mentioned methods differ mainly in details about how a map of the collected peaks is built. In the ALE approach, each peak is modelled as a 3D gaussian distribution with a given standard deviation, and the probability that a given voxel contains at least one activation peak is given by the union of the probabilities for single peaks to fall into that voxel. A map reporting this "activation likelihood estimate" for each voxel is thus obtained. The AGES method is very similar, in that it uses gaussian modelling of each peak to obtain a statistic image; this is then thresholded on the basis of the results of a permutation test, and the surviving regions are taken to be the areas of consistent activation. In the KDA method a map is built where each voxel indicates the number of nearby (within a pre-specified radius) activation peaks; in its evolution, MKDA, the focus shifts from single peaks to single studies, and thus the created maps report, for each voxel, the average number of studies for which at least an activation peak falls within the given radius from the voxel. This modification allows for an improved handling of meta-analysis datasets where a large disproportion is observed among peaks contributed by single studies; in such cases, in fact, the risk exists that one, or few, studies reporting a large number of activation coordinates come to dominate the results of the meta-analysis. MKDA also introduces a weighting mechanism to take into account the quality (in terms of statistical reliability) of each study.

Alternatively, clustering-based meta-analyses have been proposed in the literature (e.g.

Jobard et al., 2003; Wager and Smith, 2003). The idea here is submit the dataset of collected activation peaks to a clustering algorithm to find groups of spatially close peaks. Each of these clusters can be taken to represent a single activation area, whose cognitive role is then discussed in the light of the characteristics of the studies that produced activations in that given area. These methods therefore require two steps to be performed: (1) a clustering step, and (2) an interpretation step. Many clustering algorithms have been described in the literature (see Section 2.2 for a brief review), and therefore different options are possible as for step (1). For instance, in (Jobard et al., 2003) a hierarchical clustering algorithm was used, whereas Wager and Smith (2003) employed a center-based approach (the Partitioning Around Medoids algorithm). Different choices can be made at step (2), as well. Jobard et al. (2003) adopted a qualitative approach, whereby each cluster[8] was inspected and its functional role determined based on the mere prevalence of peaks coming from one category of contrasts with respect to other categories. Wager and Smith (2003) employed statistical testing (χ^2 test) for each cluster, so as to determine whether the composition of a cluster showed a significant disproportion of peaks from one category.

Both families of approaches for performing a meta-analysis of neuroimaging data have their merits. Whereas kernel-based methods appear to be more suitable for creating "consistency maps" for a given cognitive task of interest, they might be less flexible when it comes to compare activation data coming from different contrasts. It is indeed possible to apply, say, the ALE approach to the difference map obtained by subtraction from the two maps corresponding to the tasks to be compared. Still, in many situations one may want to identify, within the generic neural network associated with a cognitive process, differential activations corresponding to specific subprocesses. Using a kernel-based method, this could be achieved by creating one map for each subprocess. However, this kind of analysis appears to be more naturally carried out using a clustering-based approach: coordinates coming from all possible contrasts of interest are collapsed together and clustered in groups. Then, the specificity of each cluster for each targeted cognitive subprocess can be separately investigated, without the need to perform an a priori segregation within the input dataset. This approach has the advantage to allow for interesting effects, not predicted at first, to emerge from subsequent analysis of the obtained clusters, as no information is excluded as it would happen if the activation data were partitioned and separately processed.

In closing, we mention here that the meta-analytic approaches summarized above all fall into the coordinate-based category, because they operate on a set of coordinates as reported in single activation studies. Conversely, in image-based meta-analysis, full statistic maps from published works are used (see Salimi-Khorshidi et al., 2009, for a comparative treatment of the two approaches). While working on the original statistical maps provides a more accurate source of data, in practice coordinate-based techniques are more widely used because gaining access to the full statistical maps from a study is not always possible.

[8]Not every cluster obtained from the clustering step was discussed in this paper: an a priori selection of clusters to be further examined was performed, so that only clusters falling in regions classically involved in the investigated cognitive process – i.e. single word reading – were considered.

1.5 Conclusion

In this chapter, we have presented an overview on how functional neuroimaging can be used to advance our knowledge of the mind. Of particular interest for this Thesis is the meta-analytic approach, which has the potential for offering new, and clearer, insights into the mechanisms of cognition, by building on previously published data from which the most consistent results can be extracted. In our work, we have approached the meta-analysis tool both from the methodological side, and from the application side.

In Chapter 2 we will describe our contributions for improving the clustering-based meta-analytic approach. We chose to adopt a hierarchical clustering algorithm in our work, as in Jobard et al. (2003), because it has the advantage, with respect to central clustering algorithms, that it does not require to pre-specify the number of clusters to be obtained. This can be determined after clustering has been performed, by providing constraints defined on the characteristics of the desired clusters (e.g., we want a solution where the average dispersion within the clusters is below a given threshold); the final number of clusters just follows from the application of these constraints. This is advantageous because it is often very difficult to determine what a good number of clusters should be for the problem at hand. However, a known – although not commonly appreciated – problem of hierarchical clustering is an inherent instability that can lead to different clustering solutions to the same, but differently permuted, input data. This problem, known as the problem of non-uniqueness of the solution, can make it hard to reproduce the results from a previous meta-analysis, but, most importantly, in the most extreme cases might even lead to different conclusions on the role of some clusters, for different permutations of the same data. Having a single clustering solution for each dataset seems a desirable quality here: we have therefore developed a modification of classical hierarchical clustering to solve this problem. Section 2.3 is devoted to the description of this work. Additionally, we also speculate that an additional improvement of meta-analytic methods could be expected if the clustering step made use of anatomical, "semantic" information, along with the purely spatial one. As the clustering algorithm is not aware of the existence of anatomical boundaries, such as major sulci, inconsistencies within clusters can result: for instance, a cluster can be obtained that contains activation peaks from both the cerebellum and the occipital lobe. As these areas are anatomically and functionally segregated, collapsing their activations in one cluster might not be desirable. Adding anatomical constraints to the clustering process, therefore, could help produce more anatomically plausible clusters. We discuss our efforts in this direction in Section 2.4.

As for the applications, Chapter 3 reports our work on a meta-analysis of functional neuroimaging data on single word reading. The meta-analysis was aimed at discovering specific networks, associated to subprocesses of interest, within the more generic reading network. By employing our clustering algorithm and performing statistical testing for each resulting cluster, we identified a word-related network, a pseudoword-related network, and a difficulty-sensitive network that were discussed in the light of the body of the neuroimaging literature, and of the major cognitive theories on reading. Our results therefore offer a condensed picture

of the neuroimaging of single word reading and help clarify the nature of involved subprocesses. Finally, we mention here that the methods developed during this Ph.D. project were also applied to the meta-analysis of different domains other than reading: noun and verb processing[9] (Crepaldi et al., In Preparation), aging (Berlingeri et al., In Preparation), and motor imagery (Invernizzi et al., In Preparation).

[9]It should be noted that a different approach to the problem of non-uniqueness of the solution, than the one described in Chapter 2, was used for this work.

Chapter 2

Improving the clustering-based approach to the meta-analysis of functional neuroimaging data

"There is no truth in clustering!"
— I. Cattinelli, 1981–

2.1 Introduction

In this chapter, we will describe our work aimed at improving the procedures that are currently employed in clustering-based meta-analyses of functional neuroimaging data. We focused mainly on the clustering step involved in such procedures, and in particular on the problem of non-uniqueness of the solution: in many clustering algorithms, different permutations of the input data can return different clustering solutions, and these in turns can be associated with different interpretations of the same data. In what follows, after a brief review of clustering methodologies, we will present a modification of classical hierarchical clustering developed to guarantee that a unique solution is returned for every input dataset. The materials presented in this chapter constitute the basis of a journal paper that is about to be resubmitted in revised version (Cattinelli et al., Under Revision–b).

Our algorithm has been applied, among the others, to the meta-analysis of functional neuroimaging data on single word reading: this work is reported in the next chapter. Finally, a set of other minor, yet useful, contributions to an improved meta-analytic methodology will be described in the closing of this chapter.

2.2 Clustering: a review

Discovering similarities in the real world is a fundamental task for both humans and machines, as it allows, for instance, reasoning by categories (Goldstone, 1994), a powerful instrument for handling the complexity of real-world experiences. This task can be carried out by

clustering (Xu and Wunsch, 2008, 2005; Jain et al., 1999; Cormack, 1971), which can be defined as the process of automatically grouping objects into subsets, based on their similarity. Once a similarity (or dissimilarity) measure over input objects has been defined, a clustering algorithm looks for groups (*clusters*) of objects such that those inside a cluster are most similar among them, and most dissimilar to objects belonging to other clusters. Thus, the grouping occurs based only on features of the input elements themselves: no guiding from a user is given nor required, and the clustering algorithm must discover by itself the underlying structure of the dataset. For this reason, clustering is an instance of *unsupervised learning*. A related task is *classification*, which assigns input objects to classes, but does so in a *supervised* fashion, as classes are pre-specified by the user, who also provides, for each object in the training set, the label of the class to which that object must be assigned (Duda et al., 2001).

Clustering is a widely used technique in many fields – e.g. bioinformatics (Jiang et al., 2004), document clustering (Steinbach et al., 2000), and many others) – as it organizes objects of interest into categories that can then be further analyzed in order to determine their (common) characteristics. In this way, it also helps reduce the complexity of a problem, because similar elements (all those grouped in a cluster) can be treated as just specific instances of a more general category, so that inferences can be made on such categories rather than on a single-item basis.

As Xu and Wunsch (2005) point out, a clustering task does not exhaust itself in the actual grouping procedure, but requires both data preparation, and subsequent evaluation of the obtained results. The process of feature selection consists in summarizing each input element by a set of descriptive characteristics, called *features*, which will be used as the basis for performing data partitioning: objects with similar features will then be grouped together. Thus, for instance, when clustering groups of people, some natural features to use might be age, height, sex, or ethnicity. As some choices of features may be more useful than others in inducing a good grouping of the data, and can also result in significant reduction of data dimensionality and, therefore, more efficient processing, it follows that the feature selection step often turns out to be crucial. After a clustering solution has been obtained, its quality might be investigated (*cluster validation*); for example, one may want to compare the results provided by two or more clustering algorithms, or determine whether the number of obtained clusters is optimal, or at least satisfactory, for the dataset at hand. To this end, several testing criteria (also called indexes) that give a measure of the quality of clustering solutions are available. After validating a clustering result, this still needs to be interpreted in the light of specific, application-related knowledge (e.g. established results in bioinformatics), in order to generate novel findings in the investigated field. Whereas all these steps (feature selection, cluster validation, and cluster interpretation) are certainly important for applications, and interesting on their own also from a theoretical perspective, for reasons of space we will restrain our treatment of the topic to a panoramic description of some of the main clustering techniques in the literature.

Given a dataset $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}\}$ of N elements belonging to the multi-dimensional

space $\mathbb{R}^D$, that is, $\boldsymbol{x}_j = (x_j^1, x_j^2, \ldots, x_j^D)^T$, the goal of a clustering process consists in dividing such dataset into a set of clusters $S = \{C_1, \ldots, C_K\}$, such that

1. $\bigcup_{k=1}^{K} C_k = \boldsymbol{X}$, and
2. $\forall\, k = 1, \ldots, K \;\; C_k \neq \emptyset$.

Notice that each input element $\boldsymbol{x}_j$ can be interpreted as a vector of (numerical) features (although also qualitative variables can be used in clustering – however, we will not consider them here) describing it; equivalently, $\boldsymbol{x}_j$ identifies the position of a point inside $\mathbb{R}^D$.

Most clustering algorithms operate by minimizing a *dissimilarity function* defined on pairs of input objects (or an object and a cluster prototype). A dissimilarity function $diss : \boldsymbol{X} \times \boldsymbol{X} \rightarrow \mathbb{R}$ satisfies the following properties for every $\boldsymbol{x}_j, \boldsymbol{x}_{j'} \in \boldsymbol{X}$:

$$\begin{aligned} &diss(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) \geq 0 \\ &diss(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) = diss(\boldsymbol{x}_{j'}, \boldsymbol{x}_j). \end{aligned} \tag{2.1}$$

If reflexivity:

$$diss(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) = 0 \text{ iff } \boldsymbol{x}_j = \boldsymbol{x}_{j'} \tag{2.2}$$

and the triangle inequality:

$$diss(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) \leq diss(\boldsymbol{x}_j, \boldsymbol{x}_{j''}) + diss(\boldsymbol{x}_{j''}, \boldsymbol{x}_{j'}) \; \forall \boldsymbol{x}_j, \boldsymbol{x}_{j'}, \boldsymbol{x}_{j''} \in \boldsymbol{X} \tag{2.3}$$

also hold, then *diss* is called a metric. Many choices for *diss* have been proposed in the literature (for a review, see Cormack, 1971; Xu and Wunsch, 2005); each one has different characteristics that make it suitable to a different class of problems. For instance, the use of the Euclidean distance (or l_2 norm), one of the most commonly used dissimilarity function, results in approximately hyperspherical clusters, and should thus be employed in those domains for which this is a reasonable assumption about cluster form.

Thus, different clustering algorithms may differ in their choice of the dissimilarity function. However, clustering algorithms also differ in more fundamental aspects, regarding the very principles that guide the definition and construction of clusters. According to these differences, several families of clustering approaches can be distinguished.

A first, important distinction is the one between hard and fuzzy clustering: if property

$$\forall i, k = 1, \ldots, K, \; i \neq k, \; C_i \cap C_k = \emptyset \tag{2.4}$$

holds, then *hard clustering* is obtained: an input element $\boldsymbol{x}_j$ must belong to one and only one cluster. In other words, the set of clusters S must be a partition of $\boldsymbol{X}$. When an element can belong to more than one cluster at the same time, we speak of *fuzzy clustering*.

Mostly independently from this first subdivision, several families of clustering approaches

can be found in the literature (Xu and Wunsch, 2005, 2008). Among these:

- *Center-based clustering*: a set of cluster prototypes is positioned and moved inside the data space as clusters are iteratively built by attracting input elements to their closest prototypes. Classical algorithms in this family are the *K-means algorithm* (Forgy, 1965; MacQueen, 1967), and its fuzzy counterpart, *fuzzy c-means* (Bezdek, 1981).
- *Hierarchical clustering*: a hierarchy of clustering solutions is built by iterative operations, either merging or splitting, on pairs of objects (input elements, or existing clusters), according to a given similarity measure (Murtagh, 1983).
- *Neural network-based clustering*: clusters are represented by output units of an artificial neural network; prototypes take the form of weight vectors. Output units compete to "win" the presented input pattern – the winner is the unit whose weight vector is more similar to the input pattern – and the weight vector of the winning unit (and possibly the vectors in its neighborhood) is moved to be closer to that pattern. Popular approaches in this family are the SOFMs (Self Organizing Feature Maps) (Kohonen, 1982, 1990), and Neural-Gas (Martinetz et al., 1993).
- *Kernel-based clustering*: input patterns are mapped into a higher-dimensional feature space, where clustering is carried out, so that a nonlinear separation problem can be simply turned into a linear one (e.g. Schölkopf et al., 1998; Schölkopf and Smola, 2002). A related approach is spectral clustering (see for a review von Luxburg, 2007), which appears particularly suitable for those domains where clusters can assume virtually unconstrained forms (such as circles or spirals).
- *Mixture densities-based clustering*: input data points are regarded as samples taken from a number of probability distributions, whose parameters have to be estimated (see, for instance, Duda et al., 2001).
- *Graph theory-based clustering*: input data points are represented as nodes in a graph, and their (dis)similarity is represented by edges (Harary, 1969; Jain and Dubes, 1988).
- *Search techniques-based clustering*: they explore the space of possible partitions in search for the optimal clustering solution (that is, the partition for which the defined cost function is optimized) by using, among others, stochastic optimization techniques (e.g. genetic algorithms, Holland, 1975).

Note that these subdivisions need not be mutually exclusive – for instance, a kernel-based approach is generally also a center-based approach. The vastness of the literature on clustering makes it hard to provide here an in-depth discussion for all, or even most, of these families of clustering. Given their wide popularity in many application domains, we decided to focus on providing an overview of (hard) center-based and hierarchical clustering. The main algorithms in the fuzzy clustering family will also be presented.

2.2.1 Center-based clustering

In classical center-based clustering, an objective function is given that has to be optimized via an iterative process. It is common for such function to be a measure of how close input data currently are with respect to the cluster prototypes: prototypes attract nearby input data points, so that each point is assigned to the cluster whose prototype is the closest one to the point, according to the chosen dissimilarity measure.

Probably the most popular algorithm in this family is the *K-means* algorithm (Forgy, 1965; MacQueen, 1967): here clusters are built so that the sum of squared errors for the clustering solution S

$$J(S, \boldsymbol{\mu}) = \sum_{k=1}^{K} \sum_{\boldsymbol{x}_j \in C_k} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_k \right\|^2 \tag{2.5}$$

is minimized. K is the number of desired clusters, and $\boldsymbol{\mu}_k$ is the prototype for cluster C_k; in the K-means algorithm, prototypes are the cluster centroids. It is evident from Eq. 2.5 that the adopted dissimilarity measure, $diss(\boldsymbol{x}_j, \boldsymbol{\mu}_k)$, is the squared Euclidean distance.

A crucial step consists in carefully initializing the K prototypes for the clusters being built. The algorithm starts by giving an initial set $\boldsymbol{\mu^0} = \{\boldsymbol{\mu_1^0}, \ldots, \boldsymbol{\mu_K^0}\}$ of cluster centroids. Elements in the input dataset are assigned to the clusters as to minimize Eq. 2.5: each point is moved to the cluster whose centroid is the closest one to the point. Cluster centroids are then updated to reflect the new cluster composition; in fact, as a result of the performed assignments, centroids have been moved to new positions: $\boldsymbol{\mu^1} = \{\boldsymbol{\mu_1^1}, \ldots, \boldsymbol{\mu_K^1}\}$. These two steps (data assignment and centroid update) are iteratively performed until the algorithm has converged, that is, no point is moved from its current cluster anymore. Such set of clusters is the final solution returned by the K-means algorithm.

Centroids can be updated in an on-line fashion or in batch mode: in the former case, a cluster centroid is recomputed immediately after an input element is moved to that cluster; in the latter case, a complete sweep of the dataset is performed before all cluster centroids are updated. Batch update is considered to be more stable and less input order-sensitive than on-line methods.

One drawback of the K-means approach lies in the fact that the desired number of clusters must be specified a priori (exactly K prototypes must be provided, and their number cannot grow nor shrink during the clustering process). This can represent a problem as it is not always clear what the ideal number of clusters for a dataset should be. This problem has been addressed, among others, by (Ball and Hall, 1967), with the ISODATA (Iterative Self-Organizing Data) algorithm, where existing centroids can be split, or merged, in a way as to encourage a good coverage of the input space. It also addresses another problem of the K-means approach, namely its sensitivity to outliers, by discarding clusters that have low cardinality. The PAM (Partitioning Around Medoids) algorithm (Kaufman and Rousseeuw, 1990) proposes a different solution, by using *medoids* as cluster prototypes: the medoid for a cluster is the data point belonging to it that has minimal average distance from all other points

in that cluster. This alleviates the outlier sensitivity problem, since this mainly depends on the fact that centroids, because they average across data points, outliers included, are used as cluster prototypes.

2.2.2 Hierarchical clustering

In hierarchical clustering (from now on, HC), not a single solution, but a hierarchy of solutions is obtained. The solution at level l of the hierarchy can be regarded as a *refinement* of the solution at level $l-1$. Hierarchical clustering can be further divided into agglomerative and divisive approaches. In *agglomerative HC*, at each processing step (i.e., at each level of the hierarchy) two existing clusters are merged into one; in *divisive HC*, at each step one cluster is split into two new clusters. Agglomerative approaches are more commonly used (especially because they are less computationally demanding than divisive ones), and therefore here we will focus exclusively on them.

Agglomerative HC partitions the data as follows. At start, each element is assigned to a different cluster (partition S_1). At each step, two clusters are merged, and a new data partition is generated. The procedure is repeated iteratively until a partition containing a single cluster is obtained (S_N). The result is a hierarchy of nested clustering solutions (i.e., partitions of the data), $T = \{S_1, S_2, \ldots, S_N\}$, where S_m is the clustering solution obtained after m steps and it is constituted of $N - m + 1$ clusters. The hierarchy of partitions can be represented in a tree-like structure, called *dendrogram* (Fig. 2.1).

The clusters composing the final solution are obtained by cutting the dendrogram at some level, according to a user-defined figure of merit like, for instance, the number of desired clusters or maximum average intra-cluster variance. The cut can be performed by climbing up the dendrogram, starting from the leaves, and stopping just before the figure of merit exceeds the defined threshold.

The two clusters being merged at each step are those that are closest to each other, according to the selected dissimilarity measure. Notice that we are considering a measure of distance between *clusters*, rather than single data points. We therefore need to introduce a dissimilarity measure between clusters, that we denote here as $Diss_C(.,.)$. Many dissimilarity measures can be chosen to be employed, each resulting in a different HC algorithm (see for instance Cormack, 1971):

1. Single linkage:

$$Diss_C(C_i, C_j) = \min_{\boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j} diss(\boldsymbol{x}, \boldsymbol{y}) \tag{2.6}$$

2. Complete linkage:

$$Diss_C(C_i, C_j) = \max_{\boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j} diss(\boldsymbol{x}, \boldsymbol{y}) \tag{2.7}$$

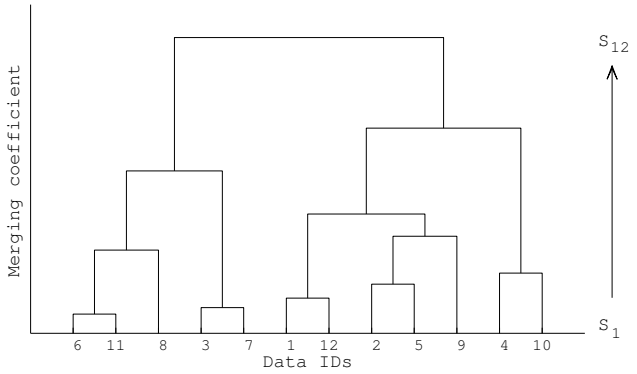


Figure 2.1: At start, each input element (IDs on the x-axis) is assigned to a singleton cluster (partition S_1). At each step, the two closest clusters are merged, decreasing the number of clusters by one. At the last step, only one large cluster is obtained (partition S_{12}). The sequence of merging steps is represented in a tree structure, called a dendrogram. The height of the horizontal segments representing merging steps is the dissimilarity value of the two clusters being merged. The dendrogram is then cut at the desired level to get the final clustering solution.

3. (Weighted) group average linkage:

$$Diss_C(C_i, C_j) = \frac{\sum_{\boldsymbol{x}\in C_i}\sum_{\boldsymbol{y}\in C_j} w_x w_y diss(\boldsymbol{x},\boldsymbol{y})}{\sum_{\boldsymbol{x}\in C_i}\sum_{\boldsymbol{y}\in C_j} w_x w_y}; \tag{2.8}$$

in (unweighted) group average linkage, $w_x = w_y = 1$; otherwise, different weights are used that depends on the "merging" history of the involved points.

4. Centroid linkage:

$$Diss_C(C_i, C_j) = diss(\boldsymbol{\mu_i}, \boldsymbol{\mu_j}), \tag{2.9}$$

where $\boldsymbol{\mu_i}$ is the centroid of cluster C_i.

5. Median linkage:

$$Diss_C(C_i, C_j) = diss(\boldsymbol{cen_i}, \boldsymbol{cen_j}), \tag{2.10}$$

where $\boldsymbol{cen_i}$ is the center of cluster C_i, computed as the average of the centers of the clusters composing C_i.

6. Ward's method (Ward, 1963):

$$Diss_C(C_i, C_j) = \Delta ESS_{i,j} = ESS_{i,j} - ESS_i - ESS_j \tag{2.11}$$

that is, the increase in the total error sum of squares (ESS) resulting from merging C_i

and C_j:

$$\begin{aligned} ESS &= \sum_{k=1}^{|C|} ESS_k \\ ESS_k &= \sum_{\boldsymbol{x} \in C_k} \|\boldsymbol{x} - \boldsymbol{\mu_k}\|^2 \end{aligned} \tag{2.12}$$

where $|C|$ is the number of clusters, and $\boldsymbol{\mu_k}$ is the centroid of cluster C_k. Thus, in Ward's method the dissimilarity between two clusters is a measure of their (collective) variance. As a result, each solution S_m in the final hierarchy is an approximation to the m-partition of the input data having minimum total intra-cluster variance.

Algorithms 1–3 are collectively called *graph methods*: they take into account every input element belonging to two clusters when computing their dissimilarity value. On the contrary, algorithms 4–6 are called *geometric methods* as they use only information about cluster centers to compute dissimilarity values.

The values of $Diss_C$ are stored in a matrix H, called dissimilarity matrix; at each step, the pair of clusters with the minimum dissimilarity value is merged into a new cluster, and the dissimilarity value between this new cluster and any other existent cluster is computed. The dissimilarity value for the merged clusters is referred to as *merging coefficient* for that time step. The update of H can be conveniently carried out by employing the Lance-Williams formula (Lance and Williams, 1966, 1967):

$$\begin{aligned} Diss_C(C_k, \{C_i, C_j\}) = \alpha_i Diss_C(C_k, C_i) + \alpha_j Diss_C(C_k, C_j) + \beta Diss_C(C_i, C_j) + \\ + \gamma \left| Diss_C(C_k, C_i) - Diss_C(C_k, C_j) \right| \end{aligned} \tag{2.13}$$

where C_i and C_j are the two clusters joined to form the new cluster, and C_k is any other cluster ($k \neq i, j$). Different values of $\alpha_i, \alpha_j, \beta$, and γ are associated with different HC methods. Table 2.1 reports the coefficients associated with the dissimilarity measures introduced above. For Ward's method, it can be proved (see Section A.1 of the Appendix) that, if $diss(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2$, then applying Eq. 2.13 with the corresponding coefficients, as given in Table 2.1, actually yields $Diss_C(C_i, C_j) = 2\Delta ESS_{i,j}$.

We have already mentioned that the final step in the application of a HC algorithm is the cut of the dendrogram, that is, deciding which level of the returned hierarchy of solutions must be selected as the final clustering. As the user can give constraints about features of the final clusters (in the form of a threshold on the average intra-cluster dispersion, for example), the dendrogram can be climbed up until these constraints are violated, and the solution at the immediately previous level can be returned. This approach has the advantage to provide a good solution (in the sense of a user-defined criterion) without having to specify a priori the number of clusters. This, along with the independence from initialization (there is no

HC Algorithm	α_i	α_j	β	γ
Single Linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete Linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Group Average Linkage	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Weighted Group Average Linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	0
Centroid Linkage	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$-\frac{n_i n_j}{n_i+n_j^2}$	0
Median Linkage	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0
Ward's Linkage	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{n_j+n_k}{n_i+n_j+n_k}$	$-\frac{n_k}{n_i+n_j+n_k}$	0

Table 2.1: Coefficients for the Lance-Williams formula corresponding to different HC algorithms (adapted from Cormack, 1971).

initial set of centroids to be provided), constitute the major strengths of HC approaches versus center-based ones.

Conversely, HC is generally more computationally demanding ($O(N^2)$), in both time and space) than center-based algorithms (for instance, the K-means algorithm is approximately linear in time). It also lacks robustness, in that possible misclassifications cannot be corrected: once an input element has been assigned to a cluster, it cannot migrate to a different one. More efficient and robust algorithms have been proposed in recent years: for instance, BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies, Zhang et al., 1996) can handle large datasets by using optimized data structures (a height-balanced tree whose vertexes represent information about clusters in a compact way), achieving a complexity of $O(N)$, while also introducing mechanisms for outlier removal. Lastly, we briefly mention that some HC approaches – namely, median and centroid linkage – are subject to the inversion (or reversal) phenomenon (Morgan and Ray, 1995), due to the fact that the sequence of merging coefficients can be non-monotonic when adopting these dissimilarity measures. In these cases, a distorted dendrogram is produced, where later merging steps produce partitions whose clusters are actually more homogeneous than those in lower levels of the hierarchy, which complicates the interpretation of the hierarchy itself.

2.2.3 Fuzzy clustering

As we mentioned earlier, when property 2.4 does not hold, fuzzy clustering is obtained. In other words, an input element $\boldsymbol{x}$ can belong to more than one cluster (actually, to all of them) with different degrees of memberships. At the end of the clustering process, each input element will be characterized by a membership vector, collecting degrees of membership to each cluster. This can be especially useful in those application domains where clusters are not expected to be sharply separated and some ambiguity about their boundaries is assumed, as it is common in natural language-defined categories. Fuzzy clustering was originally introduced by Ruspini (1969), based on the theory of fuzzy logic (Zadeh, 1965). Before proceeding to

describe how fuzzy clustering works, we will briefly summarize the main ideas behind fuzzy logic.

Fuzzy logic and fuzzy controllers

Very briefly, fuzzy logic can be regarded as an extension of classical boolean logic, where truth values are no longer restricted to be in $\{0, 1\}$, but can take any value in $[0, 1]$. For instance, in fuzzy logic a person p is not either *tall* ($p \in T$) or *not tall* ($p \notin T$); on the contrary, p could be *rather tall* ($m_T(p) = 0.7$). Here, we indicate with the notation $m_T : \boldsymbol{X} \rightarrow [0, 1]$ the membership function for the fuzzy set T defined over elements in $\boldsymbol{X}$. A fuzzy set is therefore a collection of elements, each having a degree of membership to this set ranging from 0 to 1; an element $\boldsymbol{x}$ can belong to more than one fuzzy set, with different degrees of membership.

Ideas from fuzzy logic have been used to implement controllers for a variety of tasks (for instance, in home appliances like washing machines). Fuzzy logic is in fact flexible enough to capture and describe vague information, turning it into linguistic variables (rather than strictly mathematical ones) that best reflect human knowledge about the task. In a fuzzy system, linguistic variables are organized in a set of rules of the type IF-THEN. For instance, in a controller for a vehicle we could find a rule such as: IF *visibility is low* THEN *speed is very low*. The actual value of the variable visibility, *vis*, will trigger multiple rules, each one at a different degree; this corresponds to the degree of membership of *vis* to the fuzzy set representing the rule *premise* (e.g. if $m_{\text{LOW\_VIS}}(vis) = 0.8$, then the above rule will be activated with degree 0.8.). In turns, this determines how strongly the rule consequent will contribute to the output: the fuzzy set VERY_LOW_SPEED representing the consequent is therefore cut at 0.8. Consequents of triggered rules are then combined; the process of *defuzzification* (for instance, by computing the centroid of the combined consequents) turns these clipped fuzzy sets into a final numerical value, in our example a value of speed. Note that a rule can have more premises, combined by operators AND and OR; for instance, a rule may be that IF *visibility is high* AND *traffic is low* THEN *speed is high*. In this case, this rule will be triggered with a degree which is the minimum of the degrees of membership of the input values *vis* and $traf$ to the two premises (that is, $\min\{m_{\text{HIGH\_VIS}}(vis), m_{\text{LOW\_TRAF}}(traf)\}$); this follows directly from how operators on fuzzy sets are defined (in this case, we adopted the definition of the fuzzy AND operator by Zadeh, although other choices – e.g. Lukasiewicz operators – are possible).

Fuzzy and possibilistic clustering

As "crisp" clusters are analogous to classical sets (more precisely, to subsets forming a partition of a set), clusters produced by a fuzzy clustering algorithm are analogous to fuzzy sets. In fuzzy clustering (Ruspini, 1969) each input data point $\boldsymbol{x}_j$ is characterized by a membership vector of size K (where K is the number of clusters), $\boldsymbol{U}_j = [u_{1j}, \ldots, u_{Kj}]$, such that

$$\sum_{i=1}^{K} u_{ij} = 1 \text{ for each } j \tag{2.14}$$

and

$$\sum_{j=1}^{N} u_{ij} < N \text{ for each } i. \tag{2.15}$$

u_{ij} is the degree of membership of element $\boldsymbol{x_j}$ to cluster C_i. We call the $K \times N$ matrix $\boldsymbol{U}$, collecting vectors $\boldsymbol{U_j}$ for each j, a *fuzzy partition matrix*.

Probably the most popular fuzzy clustering algorithm is *fuzzy c-means* (Bezdek, 1981), which is the fuzzy equivalent of the K-means algorithm. In fact, as in K-means, fuzzy c-means aims at finding a (fuzzy) partition over the dataset $\boldsymbol{X}$ such that an objective function is minimized:

$$J(\boldsymbol{U}, \boldsymbol{\mu}) = \sum_{i=1}^{K} \sum_{j=1}^{N} (u_{ij})^f diss(\boldsymbol{x_j}, \boldsymbol{\mu_i}) \tag{2.16}$$

where $f \in [1, \infty)$ is the *fuzzification* parameter (the higher f, the fuzzier the resulting clusters), and *diss* is, as usual, the dissimilarity measure between $\boldsymbol{x_j}$ and the prototype for cluster C_i; in the fuzzy c-means algorithm, the squared Euclidean distance is used. Following the conventional notation, in what follows we will denote with D_{ij}^2 the (squared) distance between $\boldsymbol{x_j}$ and the prototype for cluster C_i.

As in K-means, the clustering process consists of both assigning input data points to clusters (in this case, computing degrees of membership for each data point, to each cluster) and updating prototypes, which are initially randomly selected. The membership values for an input element $\boldsymbol{x_j}$ are updated as follows. Let $I_j = \{i | \boldsymbol{x_j} = \boldsymbol{\mu_i}\}$ be the set of cluster indexes such that $\boldsymbol{x_j}$ is coincident with the prototypes of these clusters. If $I_j \neq \emptyset$, then the new u_{ij} will be $1/|I_j|$ if $i \in I_j$, 0 otherwise. If $I_j = \emptyset$,

$$u_{ij} = \frac{1}{\sum_{l=1}^{K} \left(\frac{D_{lj}}{D_{ij}} \right)^{2/(1-f)}} \tag{2.17}$$

Then, prototypes are updated according to

$$\boldsymbol{\mu_i} = \frac{\sum_{j=1}^{N} (u_{ij})^f \, \boldsymbol{x_j}}{\sum_{j=1}^{N} (u_{ij})^f} \tag{2.18}$$

Thus, the prototype for a cluster is computed as the weighted mean of its objects, where the weight is given by their degree of membership to that cluster. The described iterative process is repeated until prototypes are not appreciably moved anymore.

Being a center-based algorithm, fuzzy c-means is sensitive to the initial displacement of prototypes; some solutions have been proposed, in order to achieve a good initialization (see

for instance Yager and Filev, 1994). Another drawback of this technique lies in its sensitivity to noise and outliers. An interesting solution, actually representing an evolution of the general concept of fuzzy clustering, is *possibilistic clustering* (Krishnapuram and Keller, 1993).

In possibilistic clustering, the constraint expressed in Eq. 2.14 is relaxed, and becomes

$$\max_i u_{ij} > 0 \text{ for each } j \tag{2.19}$$

In other words, it is only required that each data point belongs with a non-null degree of membership to at least one cluster. In this approach, the value u_{ij} is interpreted as a measure of the compatibility of $\boldsymbol{x_j}$ to the prototype for cluster C_i. A new objective function is provided that has to be minimized:

$$J(\boldsymbol{U}, \boldsymbol{\mu}) = \sum_{i=1}^{K}\sum_{j=1}^{N}(u_{ij})^f D_{ij}^2 + \sum_{i=1}^{K}\eta_i \sum_{j=1}^{N}(1-u_{ij})^f \tag{2.20}$$

This is basically the same as Eq. 2.16, but with an additional term introduced to encourage the membership values to grow large. The $\eta_i > 0$ values determine the relative weight of this term within Eq. 2.20. According to this objective function, the new membership values will be determined as

$$u_{ij} = \frac{1}{1 + \left(\frac{D_{ij}^2}{\eta_i}\right)^{1/(f-1)}} \tag{2.21}$$

It can be noticed that, differently from how membership values are updated in fuzzy clustering, in possibilistic clustering the compatibility of an element $\boldsymbol{x_j}$ to a cluster C_i depends on its distance from the prototype of C_i only; distances from other prototypes are not taken into account. This is the reason why possibilistic clustering is less sensitive to noise and outliers: the compatibility of an input data point with a cluster is independent of the nearby presence of other clusters. As in the fuzzy c-means algorithm, the prototype update step follows. If the used $diss(.,.)$ function is the squared Euclidean distance, the prototypes are the cluster centroids and *possibilistic c-means* algorithm is obtained; other distance measure can be used, however, thus leading to different possibilistic clustering algorithms.

Krishnapuram and Keller (1993) also suggest how to initialize constants η_i, for instance by setting

$$\eta_i = Q\frac{\sum_{j=1}^{N}(u_{ij})^f \; D_{ij}^2}{\sum_{j=1}^{N}(u_{ij})^f} \tag{2.22}$$

where Q is constant, usually taken to be 1. This formula assigns greater weights (corresponding to a greater "bandwidth", to use the words of the authors) to dispersed clusters (that is,

those having large average intra-cluster distance).

We conclude this brief overview on fuzzy clustering techniques by remarking that a crisp partitioning of the dataset can be easily derived from fuzzy clusters, by simply assigning each data point $\boldsymbol{x}_j$ to the cluster C_i such that u_{ij} is maximum.

2.2.4 Some issues in clustering

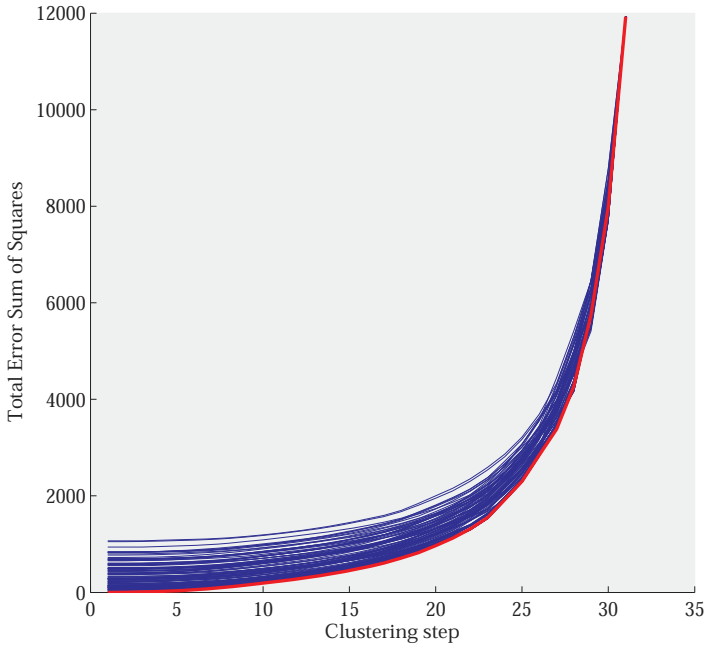
Notwithstanding the wide popularity in many disciplines, and the practical value of clustering, there are a number of issues concerning this paradigm that are worth being discussed here.

One of such issues concerns the optimality of the returned solution, which, generally, cannot be assured. In center-based clustering, it cannot be guaranteed that the final solution is the partitioning of the input data for which the cost function (e.g. Eq. 2.5, if K-means is used) is (globally) minimum. In principle, the optimal solution could be found by exhaustive enumeration of all possible partitions of the data, but this gets easily unmanageable even for relatively small datasets. Only stochastic optimization (Kirkpatrick et al., 1983), which is itself extremely costly, or a careful initialization of the prototypes allows escaping local minima. Although few attempts have been proposed to derive a robust initialization (e.g. Ferrari et al., 2007) there is no universal and reliable method for doing this, and some prototypes, during the clustering process, get stuck. These prototypes are referred to as "dead units" (Fritzke, 1995) and hamper the clustering process and the quality of the result.

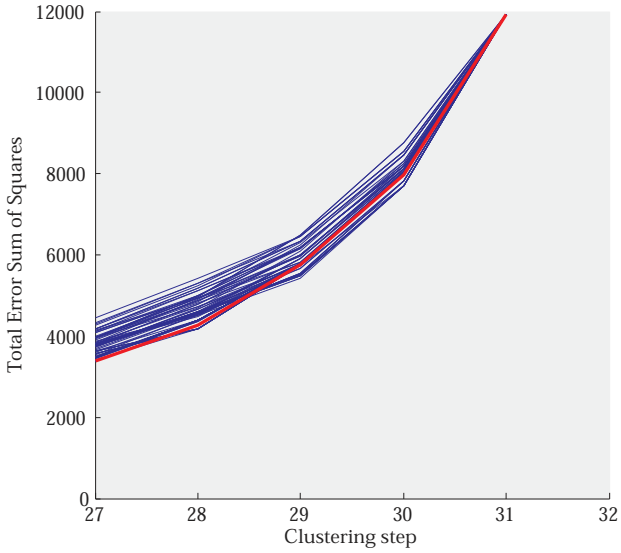
Similarly, in (agglomerative) HC, although each processing step is locally optimal as the pair of elements to be merged is chosen to minimize a dissimilarity function, the global optimality of the clustering solution cannot be guaranteed: in fact, *"even an exact algorithm for a properly defined method is not necessarily optimal. [...] Unless the clusters can be shown to have properties approximating to some desiderata for the clusters, the fact that they have been obtained by successive steps each of which was best of the steps then available seems irrelevant"* (Cormack, 1971, p. 330).

Let us consider, for instance, one of the HC algorithms that we introduced in this section: Ward's method. At each merging step, the increase in the total ESS (Eq. 2.12) of the clustering solution for that level of the hierarchy, is minimized. It could be expected that, in so doing, we are building an optimal hierarchy, that is, each solution S_m composing the final dendrogram is the $(N - m + 1)$-partition of the input dataset for which the total ESS is minimum. However, this is not the case, as each S_m can be regarded as only an approximation to the optimal partition. Figure 2.2 illustrates this point with a practical example: here, a dataset of 32 3D-points[1] was submitted to Ward's algorithm, and the ESS for each solution in the final dendrogram was computed (plotted in red in the figure). Then, we produced 100 alternative dendrograms for the same dataset, by randomly selecting the pair of objects to merge at the first clustering step, while the subsequent steps were performed in the standard way, by

[1] These coordinates constituted two of the clusters obtained in our meta-analysis of functional neuroimaging data on single word reading – see farther in this chapter, and Chapter 3.



(a)



(b)

Figure 2.2: An example illustrating the issue of non-optimality in Ward's HC method. A dataset of 32 3D points was clustered, and the ESS for each solution in the final hierarchy is plotted (red curve). 100 random solutions were also generated for the same dataset: although we expect these curves to always lie on top of the red one, since this was obtained by locally optimal merging choices, this is not always the case, as zooming in on later clustering steps (panel (b)) reveals.

choosing the pair of clusters having minimum dissimilarity value. ESS values for each solution in each dendrogram are plotted in blue in Fig. 2.2(a). We could expect that the red curve always lies below all the blue curves – that is, that the red curve corresponds to an optimal hierarchy of solutions. This appears to be the case, but if we zoom in on later merging steps (Fig. 2.2(b)) we can see that some blue curves actually get better than the red one on these partitions. This is perhaps not surprising if we think that each solution in the hierarchy is built on top of the one at the previous level, and no re-assignments are allowed: however, the optimal 2-partition of a dataset – for example – is not necessarily obtained by merging either two clusters in the optimal 3-partition, as alternative subdivisions might, with this cluster number, be more convenient.

In general, even determining what an "optimal" solution for a clustering problem is, can be less than straightforward. In the K-means algorithm, this would be, as we said above, the K-partitioning of the input dataset that minimizes the objective function. A similar definition of optimality can be given for Ward's HC approach, as we have just discussed. However, the same cannot be said for, for instance, single linkage: what is, in this case, the global objective function that we are trying to optimize? In fact, it is not uncommon for clustering algorithms not to explicitly define, and optimize, an objective function. For instance, Self-Organizing Maps (Kohonen, 1982) do not specify any global cost function, although each update step can be shown to minimize a set of potential functions, one for each prototype (Erwin et al., 1992).

In fact, difficulties in defining optimality for clustering problems are part of a more general problem of this field, namely, the lack of solid theoretical foundations. This makes it hard to even define what a clustering function is, or at least should be. Moreover, as the ground truth for a given, non-trivial dataset is usually not available[2], the evaluation of alternative clustering solutions is somewhat arbitrary, possibly based on a set of quality indexes that, however, may or may not be adequate for the clustering problem at hand (Von Luxburg and Ben-David, 2005); the choice itself of which clustering algorithm one should use for their dataset is not provided any guidance from theoretical considerations, and its aggravated by the multitude of algorithms that have been introduced in the literature along the years. In a way, clustering is still more an art than a science[3].

However, some attempts have been made to formalize the notion of clustering. Kleinberg (2003) showed that when trying to characterize a function $f : diss(.,.) \rightarrow P(\boldsymbol{X})$ (with $P(\boldsymbol{X})$ being the set of all possible partitions over $\boldsymbol{X}$) as being a clustering function if and only if it satisfies three reasonable axioms, then an impossibility result is obtained, meaning that no such function actually exists. The axiomatic properties suggested by Kleinberg (2003) were:

1. scale-invariance: the same clustering result must be obtained if the scale over which dissimilarities are defined is changed;
2. richness: all possible partitions of the input dataset must be obtained, provided that an

[2]Cf. the opening quote for this chapter.

[3]As witnessed by a NIPS 2009 Workshop entitled "Clustering: Science or Art? Towards Principled Approaches".

adequate dissimilarity measure is defined over it;

3. consistency: if dissimilarities for points inside a cluster are decreased, and those for points belonging to different clusters are increased, the same partition as the one obtained with the original dissimilarity values must result.

Although no function can satisfy these three axioms at the same time, popular existing clustering methods were shown to embody two of these axioms. For instance, Kleinberg (2003) considered single linkage, coupled with three stopping conditions[4] (in our terminology, these correspond to different ways of cutting the dendrogram to obtain the final clustering solution). If HC is stopped when K clusters are built, then richness is violated (partitions consisting of a different number of clusters cannot be obtained), but scale-invariance and consistency are satisfied; if the merging process is stopped when the merging coefficient is larger than an absolute threshold, then scale-dependence is introduced, and only richness and consistency are retained; lastly, if a relative stopping threshold is used (a proportion of the maximum merging coefficient), consistency is lost, but richness and scale-invariance are satisfied. Similarly, K-means was shown to violate consistency. This analysis suggested that it should be possible to characterize clustering algorithms based on the set of properties (and possibly relaxations of them) they do/do not satisfy.

Note that Kleinberg's impossibility theorem is not necessarily a proof that clustering cannot be axiomatized at all. In fact, different axioms may be used (for instance, the relaxed versions of the original axioms given above), for which there exist some functions that simultaneously satisfy all of them. This was done in (Bosagh Zadeh and Ben-David, 2009), by relaxing the original consistency and richness axioms so that they are now restricted to fixed size (K) partitions only. Bosagh Zadeh and Ben-David (2009) also introduced an order-consistency axiom, whereby it is required that the same K-clustering is obtained for two dissimilarities measures, if the ordered list of pairs of input points is the same for both of them (in our terminology, the ordering of pairs would result from sorting the initial dissimilarity matrix in ascending order). Given this set of four axioms, it was shown that they are consistent, as single linkage does satisfy all of them. A uniqueness result was also derived showing that single linkage is the only clustering function that satisfies the axioms and the additional property of MST-coherence: that is, if two dissimilarity functions over the same dataset induce the same minimum spanning tree of the linkage graph (see note 4), then this property states that the resulting clusterings must be the same. In (Ackerman et al., 2010) a further step was made, by providing a set of properties that uniquely identify all linkage-based algorithms. These are: the hierarchical nature of merging operations; outer-consistency (if two dissimilarity measures are considered, one of which is larger than the other for points belonging to different clusters, but are the same for points inside a cluster, then the clusterings obtained using these

[4] In (Kleinberg, 2003; Bosagh Zadeh and Ben-David, 2009; Ackerman and Ben-David, 2009), single linkage is presented as a graph theory-based algorithm: nodes represent points in the input dataset, and edges reflect their dissimilarity values. A clustering is built by progressively adding edges to the initially disconnected graph, starting from the smallest edges (smallest dissimilarities), until a stopping criterion is met. Then, connected components in the graph are taken to be the clusters for the input dataset.

two measures are identical); extended richness (if disjoint groups of points are given, it is always possible to set a dissimilarity measure such that each group ends up to be in its own cluster); and locality (if a subset of K clusters obtained with a clustering function is submitted again to the same function asking for K clusters, the same clusters are returned).

Ackerman and Ben-David (2009) took an alternative approach to the axiomatization of clustering, by re-defining the same set of axioms proposed by Kleinberg (2003) so that they referred to clustering quality measures, rather than clustering functions; thus, for instance, the scale-invariance property now states that the quality measure for a clustering must not change if dissimilarity values are subject to a change in scale. The set of re-defined axioms is now consistent, that is, there exists some quality measures that satisfy all of them. The authors also introduced a novel axiom, isomorphism invariance, to achieve some form of completeness for their set of axioms – that is, no functions that clearly are not clustering quality measures must be able to satisfy all axioms. Isomorphism invariance requires that the quality measure for two isomorphic clusterings be the same. These theoretical works show that the same definition of what a clustering function is eludes easy formulations, although these recent attempts to isolate desirable properties for clustering, and formalize families of algorithms based on them, are showing the first, promising results.

Lastly, another issue in clustering, which is both theoretically interesting and of practical impact, is the problem of non-uniqueness of the solution. In fact, different clustering solutions can be returned, when a different ordering for the same input dataset is provided to the clustering algorithm (Sibson, 1972; Morgan and Ray, 1995). This problem *"certainly is not widely known"* (Van der Kloot et al., 2005) and it is usually disregarded in applications of clustering; however, it is not a trivial problem if we consider that, as a consequence, the actual conclusions drawn from clustering may be only the result of a particular presentation order of the input data. Few attempts have been made to solve this problem. In (Van der Kloot et al., 2005) it is suggested to run the clustering process on different permutations of the input data and to choose the solution that minimizes the defined cost function. However, it cannot be guaranteed that a different data permutation would not produce an even better solution. On the other side, an exhaustive generation and exploration of all alternative solutions associated with a dataset is computationally infeasible.

Arguably, a reasonable requirement for clustering is that the returned solution is unique. For this reason, we worked on this issue to try and produce a clustering approach that would be insensitive to permutations of the input dataset. This effort was conceived within the working framework of improving the methodology for clustering-based meta-analysis of functional neuroimaging data. For this reason, we chose to work specifically on Ward's HC method. As we mentioned above, HC has the advantage of not requiring the number of desired clusters to be pre-specified, which is a useful feature in the considered domain as this parameter cannot be easily determined by the user. We chose to adopt Ward's method, among the possible HC algorithms, because the effect of using this dissimilarity measure, along with

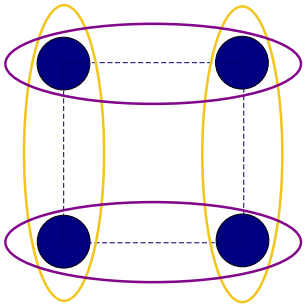


Figure 2.3: Four data points lie at the corners of a square: therefore, pairs on each side of the square have the same (Euclidean) distance, which leads to four minimal-dissimilarity (MD) pairs. If we run a HC algorithm on this dataset and cut the resulting dendrogram to get a 2-clusters solution, two different solutions are obtained (here shown in yellow and purple, respectively), according to which pair of points is selected first.

the use of the squared Euclidean distance as a measure of dissimilarity between single input data points, is to obtain compact (i.e., having low within-cluster variance), spherical clusters, which is especially desirable when clustering cerebral activation peaks.

In the next section, we propose a modification of the original HC algorithm that has been designed to solve the problem of non-uniqueness of the solution; this problem, with respect to HC approaches, is presented in Section 2.3.1. Our approach is based on generating only the subset of *significantly different* solutions, thus keeping the computational load relatively low but, at the same time, ensuring that no interesting solution is missed; to this end, we developed a novel algorithm that is based on the definition of an equivalence relation over the clustering solutions associated with a dataset. The algorithm is described in Section 2.3.2. Application of this novel method to both the neuroimaging and bioinformatics domain, presented in Section 2.3.3, provides a demonstration of the relevance and practical utility of the approach. Lastly, a discussion of the main features of the approach is offered in Section 2.3.4.

2.3 A Novel Approach to the Problem of Non-uniqueness of the Solution in Hierarchical Clustering

2.3.1 The non-uniqueness of the solution in hierarchical clustering

As we mentioned above, HC can return different solutions depending on the order in which the input data are presented (see Fig. 2.3). This is due to the presence of *ties* in the dissimilarity matrix at a given step, that is the minimum dissimilarity value, v, is shared by more than one cluster pair.

Definition 2.3.1. *Let* $v = \min_{C_i,C_j} Diss_C(C_i, C_j)$, *where* C_i *and* C_j *are clusters available at the current processing step,* t. *We call* ***minimal-dissimilarity pair*** *(MD pair) each pair of clusters* $p = (C_i, C_j)$ *such that* $p \in P_t = \arg\min_{C_i,C_j} Diss_C(C_i, C_j)$; *that is,* $Diss_C(p) = v$ *for each MD pair* p.

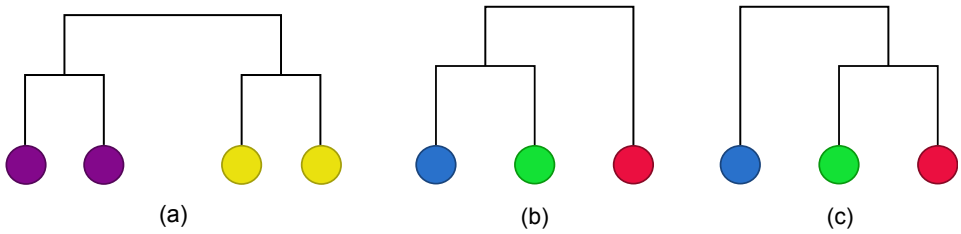


Figure 2.4: The purple and the yellow pairs in panel (a) are two non-critical pairs, that is MD pairs that have no cluster in common: whichever pair of points is merged first, the final dendrogram is the same. On the other hand, the three elements shown in panel (b) and (c) produce two critical pairs (the blue-green one, and the green-red one) from which two different dendrograms can be obtained, depending on which pair is selected first.

At each step t (i.e., every time the dissimilarity matrix is updated), we might have more than one MD pair; that is, $|P_t| > 1$. The order in which the input data points are presented to the algorithm determines the order whereby cluster pairs are found inside matrix H; current algorithms just select the first MD pair encountered when browsing H. Therefore, a different permutation of the input data points can lead to the selection of a different MD pair, and this, in turns, can produce a different dendrogram.

2.3.2 Algorithm description

The solution to the non-uniqueness problem delineated above that we propose here is based on identifying what we have called the *significantly different* alternative dendrograms that result from the selection of different MD pairs. To this end, we need to introduce the concept of *critical* pairs, and then that of *equivalent* pairs.

At each processing step, the choice of one MD pair in place of another does not necessary lead to different dendrograms; in some cases, different choices will lead to different merging sequences, but the shape of the resulting dendrograms will be the same (see Fig. 2.4a). In other words, we can often find situations where the choice between MD pairs is *not critical* as it does not produce different dendrograms. We will call such MD pairs *non-critical*.

Definition 2.3.2. *A MD pair $p = (C_i, C_j)$ is a **non-critical pair** if $\forall p' = (C_{i'}, C_{j'})$, $p' \neq p$ being a MD pair, $i \neq i', j'$ and $j \neq i', j'$ hold.*

Non-critical pairs are therefore those pairs that do not share any element with other MD pairs. The choice of merging one non-critical pair in place of another does not affect the shape of the resulting dendrogram because these choices are not mutually exclusive: the choice of a non-critical pair leaves other non-critical pairs available for subsequent merging.

This can be also seen by analyzing H. Let us suppose that H contains n_p entries that have the same MD value, v, and that these entries are distributed such that for each row and column at most one entry is equal to v (it can be shown that this is another way to state Def. 2.3.2). Whenever one MD pair, say (C_i, C_j), is merged, dissimilarity values for clusters C_i and

C_j are discarded, which corresponds to deleting the i-th row and the jth column[5]. None of the other MD pairs would be touched by such operation. Therefore, at the subsequent clustering step, one of the remaining MD pairs, with $Diss_C(.) = v$, would be selected for merging, and so on, until all the non-critical pairs with $Diss_C(.) = v$ have been merged. Since all these pairs have merging coefficient equal to v, the same dendrogram is obtained regardless of the specific merging sequence.

Put differently, the choice among non-critical pairs cannot open new scenarios where a new MD pair appears, which would make the order whereby non-critical pairs are selected relevant. This is guaranteed by the fact that:

Theorem 2.3.3. *Let v be the minimum value in the dissimilarity matrix and therefore the merging coefficient in the current clustering step; let C_i and C_j the clusters being merged. In a HC algorithm employing Ward's method, each new dissimilarity value v' for the newly created cluster $\{C_i, C_j\}$ is such that $v' \geq v$. If (C_i, C_j) is a non-critical pair, then $v' > v$.*

Proof. Let us recall that (by Eq. 2.13 and Table 2.1) the Lance-Williams updating equation for Ward's method is given as:

$$Diss_C(C_k, \{C_i, C_j\}) = \frac{n_k + n_i}{n_i + n_j + n_k} Diss_C(C_k, C_i) + \frac{n_k + n_j}{n_i + n_j + n_k} Diss_C(C_k, C_j) + \\ - \frac{n_k}{n_i + n_j + n_k} Diss_C(C_i, C_j) \tag{2.23}$$

Then, for a generic cluster C_k ($k \neq i, j$) the dissimilarity value v' of C_k from the new cluster $\{C_i, C_j\}$ is computed as

$$v' = \frac{1}{n_i + n_j + n_k} \left(z(n_k + n_i) + w(n_k + n_j) - v(n_k) \right) = \\ = \frac{1}{n_i + n_j + n_k} \left(n_k(z + w - v) + n_i z + n_j w \right)$$

where $z = Diss_C(C_k, C_i)$, $w = Diss_C(C_k, C_j)$. Since v is the minimum value in the dissimilarity matrix, and (C_i, C_j) is a non-critical pair, $z > v$ and $w > v$ hold; that is, (C_k, C_i) and (C_k, C_j) cannot be MD pairs; otherwise, (C_i, C_j) would not be non-critical by definition. Therefore we can write $z = v + \varepsilon$, $w = v + \eta$ ($\epsilon > 0$, $\eta > 0$), and

$$v' = \frac{1}{n_i + n_j + n_k} \left(n_k(v + \varepsilon + \eta) + n_i(v + \varepsilon) + n_j(v + \eta) \right) = \\ = \frac{(n_i + n_j + n_k)v}{n_i + n_j + n_k} + \frac{n_k\varepsilon + n_k\eta + n_i\varepsilon + n_j\eta}{n_i + n_j + n_k}$$

from which $v' > v$ follows. □

We notice that the case $v' = v$ can only occur when both $Diss_C(C_k, C_i)$ and $Diss_C(C_k, C_j)$

[5]We are here implicitly assuming that H is stored as a triangular matrix.

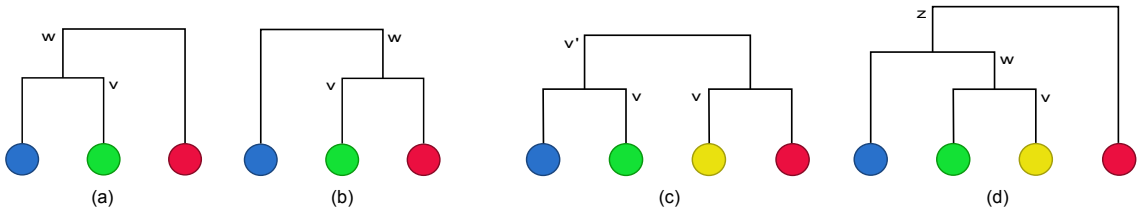


Figure 2.5: Panels (a) and (b) show two equivalent pairs, the blue-green and the green-red one. The element closest to the first pair is the blue one, and the one closest to the second pair is the red one. This guarantees that these three elements will be grouped in the same cluster. Therefore, although the two dendrograms are different, they are equivalent; notice that in intermediate steps the clusters obtained are different. Also notice that the sequence of merging coefficients (in this case, v, w) is the same, independently of which pair is first merged. The blue-green pair and the green-yellow one shown in Panels (c) and (d) are non-equivalent pairs. If the blue-green pair is merged first, the yellow element will then be merged with the red one (because it is closest to it than to the newly created cluster) and we end up with different 2-clusters solutions: {blue ∪ green} and {yellow ∪ red}, or {blue ∪ green ∪ yellow} and {red}. Also notice that the merging coefficients are different in the two cases (v, v, v', and v w, z, respectively). These two dendrograms are therefore non-equivalent and they have to be fully developed.

are equal to $v = Diss_C(C_i, C_j)$, that is when the three clusters are equidistant from each other (with dissimilarity v), but in such case they would not qualify as non-critical pairs.

Therefore, we can choose to separately develop only those dendrograms resulting from merging *critical pairs*. At each processing step all MD pairs are identified; among these, non-critical pairs are identified and merged in a random order to build one single dendrogram, while separate dendrograms are developed for each critical pairs.

With this choice the number of solutions to be handled is reduced; however, in many practical situations such reduction is not large enough (cf. Section 2.3.3). A drastic reduction in the number of developed dendrograms can be attained by introducing an equivalence relation over the dendrograms (Figs. 2.5a and 2.5b).

Definition 2.3.4. *Let $p = (C_i, C_j)$ and $p' = (C_j, C_k)$ be two critical pairs for the current processing step. We say that p and p' are* ***equivalent pairs*** *if*

$$C_k = \arg\min_{C_x} Diss_C(\{C_i, C_j\}, C_x) \tag{2.24a}$$

$$C_i = \arg\min_{C_x} Diss_C(\{C_j, C_k\}, C_x) \tag{2.24b}$$

$$Diss_C(\{C_i, C_j\}, C_k) = Diss_C(\{C_j, C_k\}, C_i). \tag{2.24c}$$

When Ward's method is used, property 2.24c directly follows from the definition of critical pair (Def. 2.3.2).

Proof. Since $p = (C_i, C_j)$ and $p' = (C_j, C_k)$ are critical pairs, $Diss_C(C_i, C_j) = Diss_C(C_j, C_k) = v$, where v is the minimum value in the current dissimilarity matrix. Then, by applying the

Lance-Williams formula for Ward's method (Eqs. 2.13 and 2.23), we get:

$$Diss_C(\{C_i, C_j\}, C_k) = \frac{(n_i + n_k)Diss_C(C_i, C_k) + (n_j + n_k)v - n_k v}{n_i + n_j + n_k} = \\ = \frac{(n_i + n_k)Diss_C(C_i, C_k) + n_j v}{n_i + n_j + n_k}$$

and

$$Diss_C(\{C_j, C_k\}, C_i) = \frac{(n_j + n_i)v + (n_k + n_i)Diss_C(C_i, C_k) - n_i v}{n_i + n_j + n_k} = \\ = \frac{(n_i + n_k)Diss_C(C_i, C_k) + n_j v}{n_i + n_j + n_k}$$

from which property 2.24c follows. □

We can re-state Def. 2.3.4 as follows. Considering the three clusters C_i, C_j, and C_k, we can refer to C_k as the excluded element when pair p is chosen, and to C_i as the excluded element when pair p' is selected. Conditions 2.24a and 2.24b state that p and p' are equivalent if the closest object to p is its excluded element, and if the same holds for p'. This means that, whichever pair we select for generating a new cluster, the next merging step involving that cluster will group it with its excluded element. That is, although the shapes of the dendrograms corresponding to p and p' temporarily diverge, they will eventually converge to the same clustering solution (Figs. 2.5a and 2.5b); if p and p' are non-equivalent the shape of their corresponding dendrograms cannot be guaranteed to converge (Figs. 2.5c and 2.5d).

Def. 2.3.4 establishes an equivalence relation over dendrograms. In particular, property 2.24c guarantees that equivalent dendrograms – those associated with equivalent pairs – have the same sequence of merging coefficients. This allows us to consider and fully develop only one representative dendrogram from each equivalence class. This drastically reduces the number of dendrograms to be fully built, making the problem computationally affordable.

Once all *non-equivalent* dendrograms have been generated, the corresponding solutions can be obtained cutting each dendrogram according to the same user-defined criterion; for instance, by setting a threshold on the average intra-cluster standard deviation, and climbing up the dendrograms until this threshold is reached. Among the obtained solutions, the best one, according to the chosen quality criterion, is identified. In the applications presented here, the between-cluster error sum of squares has been adopted. The maximization of this measure favors a better separation among clusters:

$$bESS = \sum_{k=1}^{|C|} n_k(\boldsymbol{\mu_k} - \boldsymbol{\mu_X})^2 \tag{2.25}$$

where $|C|$ is the number of clusters in the solution, n_k and $\boldsymbol{\mu_k}$ are the number of elements

and the mean of cluster C_k, and $\boldsymbol{\mu}_X$ is the grand mean of the dataset $\boldsymbol{X}$[6]. Let us remark that other user-defined measures could be employed to evaluate the different clustering solutions.

Therefore, the algorithm returns a solution that is unique, up to equivalences. It is also optimal, with respect to the desired measure of quality, among the alternative solutions that the HC algorithm would return with different orderings of the input data. The operation flow of the proposed algorithm is summarized in Fig. 2.6. The key element is the state of the clustering process, that is saved each time a new dendrogram has to be developed; specifically, the state contains the current step t, the dissimilarity matrix, the non-equivalent pairs still to be examined, the parent dendrogram from which a new one will be developed, the current merging coefficient, and additional information about current clusters (their number, cardinalities, and indexes).

2.3.3 Results

Our algorithm has been applied in two different domains: meta-analysis of neuroimaging data, and analysis of protein-protein interactions.

Meta-analyses of neuroimaging data

As illustrated in Section 1.4, HC has been used in the field of functional neuroimaging as a tool for analyzing a large set of brain activation sites as those reported in a broad collection of studies investigating different aspects of a specific cognitive function (Jobard et al., 2003). In this context, the result of a HC analysis can be used to identify groups of anatomically close activation peaks that may represent functionally meaningful brain regions, constituting specific networks of cortical and subcortical areas involved in the cognitive function of interest. We present here some of the results that we have obtained in the investigation of the networks involved in single word reading; for a full account, the reader is referred to Chapter 3. In particular, we were interested to assess whether there is evidence for areas that preferentially process real words and areas that respond specifically to pseudowords (i.e. letter strings that can be read according to the orthographic rules of the language but do not exist in the vocabulary and therefore have no meaning; e.g. *pufe*). We also looked for areas that, irrespective of lexicality, show an activation modulated by the difficulty of the material to be read (e.g. irregularly-spelled words vs. regularly-spelled words, low-frequency words vs. high-frequency words).

To this end, we scrutinized the functional imaging literature on reading since 1980s and we identified 35 studies suited for our meta-analytic study, with a total of 1176 activation peaks defined by anatomical coordinates in stereotactic space. The algorithm found 128 *significantly different* dendrograms. These were cut at the level where the average standard deviation over the clusters in any of the three directions raised above $\sigma = 7.5$; this value was set in agreement

[6]Notice that $bESS = ESS_{dataset} - ESS$, where ESS is the total within-cluster error-sum-of-squares introduced in Eq. 2.12, and with $ESS_{dataset}$ we refer to the error-sum-of-squares over the whole dataset, considered as a unique cluster.

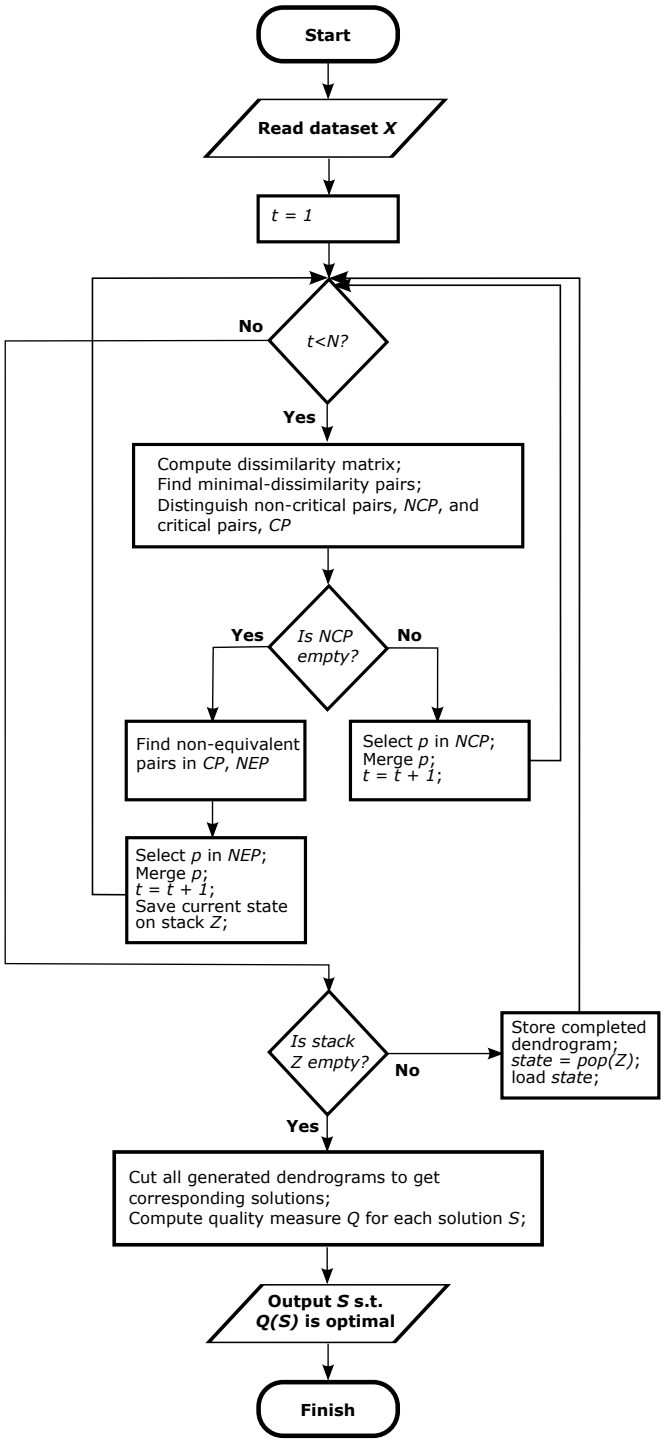


Figure 2.6: Flow chart of the proposed algorithm. To generate the set of non-equivalent dendrograms, we employ a stack structure in which the current state of the process is saved when a new dendrogram must be generated. When a dendrogram is completed, the state on top of the stack is loaded and a new dendrogram is developed starting from level t.

with (Jobard et al., 2003), to comply to the standard resolution of functional images, of about 15 mm. Four different unique solutions were identified: among these, the optimal one had a $bESS = 2.4023 \times 10^6$ (see Eq. 2.25) and consisted of 57 clusters. The other three solutions had $bESS$ equal to 2.3977×10^6, 2.4017×10^6, and 2.3980×10^6, and consisted of 55, 57, and 55 clusters, respectively. The algorithm required about 8 minutes of CPU time with MATLAB code (on an Intel Core 2 Duo Processor T72000, 2Ghz, 2 GB of memory).

The statistical analysis of the solution allowed us to identify the putative functional role of each cluster, and thus of its corresponding brain area (see Chapter 3 for details). However, as seen from Fig. 2.7, different solutions are obtained with traditional HC that, in turns, do lead to different conclusions on the role of certain brain areas. For instance, in the optimized solution two clusters, one located in the left angular gyrus and one in the left middle occipital gyrus were found (Fig. 2.7a). The former was a word-specific cluster whereas the latter fell into the non-differentiated class. However, if we considered one of the other three non-optimized alternative solutions (Fig. 2.7b) we would find that these two clusters are grouped in the same cluster, spanning over the two mentioned brain areas. Statistical testing on such larger cluster shows that it is significantly involved in word processing; therefore, in this case, the opportunity to distinguish between a more lateral region (in the angular gyrus) showing a preference for word stimuli and an occipital one that is less sensitive to lexicality, would be lost.

This example, therefore, practically shows how the problem of non-uniqueness of the solution can lead to a poor conclusion, in this case loosing spatial resolution.

HC of protein-protein interaction data

HC is one of the most used techniques in bioinformatics, with applications ranging from functional genomics, to DNA microarray data analysis, biomolecular evolution and multiple sequence alignment (Cavalli-Sforza and Edwards, 1967; Fitch and Margoliash, 1967; D'haeseleer, 2005). In several bioinformatics clustering problems data are two-valued (binary, e.g. they represent whether a given property is present or not for a given gene or protein) and are characterized by high dimensionality and sparsity. In these conditions it is likely to obtain dissimilarity matrices with ties, but the consequent problem of non-uniqueness of the solution is largely neglected.

For instance, we consider here the discovery of clusters of functionally related proteins analyzing protein-protein interaction (PPI) data. Indeed proteins that interact with the same or a similar set of other proteins are likely to share similar functions.

To reduce the computational complexity due to the high dimensionality and cardinality of PPI data, we randomly selected a random subset of 500 proteins of the model organism *S. cerevisiae* (yeast), from the original set of 5367 proteins downloaded from the *BioGRID* database (Stark et al., 2006). By applying the proposed HC algorithm, we found 96 significantly different dendrograms. The dendrograms were cut with a threshold $\sigma = 5$ on the norm of the vector representing the average standard deviation over the clusters for each dimen-

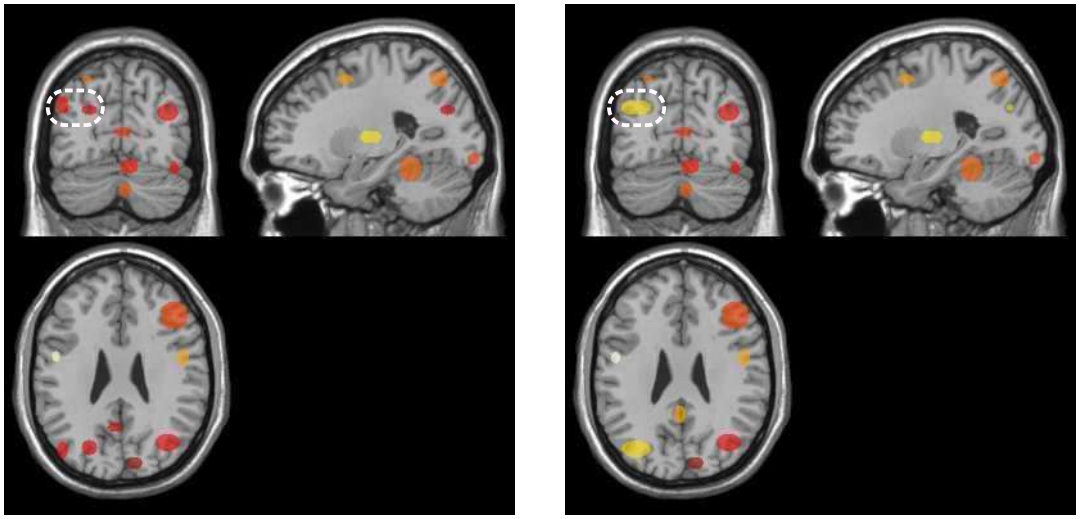


Figure 2.7: Two alternative clustering solutions for our neuroimaging dataset are shown: in panel (a), the clustering solution used for our meta-analysis ($bESS = 2.4023 \times 10^6$); in panel (b), one of the alternative solutions ($bESS = 2.3977 \times 10^6$). Each cluster is represented by a blob centered on its mean coordinate, and whose semiaxes are determined by the standard deviation of the cluster. The color of a blob codes for its cardinality. Only a section of the cerebral volume is shown. The white box highlights the difference in clustering solutions discussed in Section 2.3.3.

sion. This left us with 4 unique solutions: S_1, S_2, S_3 and S_4. The first two include 9 clusters and the last two 10. The bESS value for these four solutions was respectively 218.27, 218.40, 222.20, and 223.77. The CPU time for the whole clustering process was about 5 minutes.

To understand whether the different HC solutions lead to different biological conclusions with respect to protein-protein interactions, we performed a functional enrichment analysis of the different clusterings, a standard and widely applied technique in functional genomics (Dopazo, 2009). The main idea behind this approach consists in assigning significance to different functional categories by comparing the observed number of proteins in a specific category with the number of proteins that might appear in the same category if a selection performed from the same pool of proteins were completely random (Khatri and Draghici, 2005). We chose to adopt the Gene Ontology (GO) terms of the Biological Processes (BP) ontology as functional categories (The Gene Ontology Consortium, 2000); each GO term represents a class of genes/proteins with common functional characteristics (e.g. *catabolic process*, *response to osmotic stress* or *regulation of translation*). In our experimental context, we can thus biologically characterize each clustering through the GO terms significantly represented in its clusters.

More precisely, we applied the hypergeometric test (Agresti, 1992) to assess which GO terms are significantly overrepresented in S_1, S_2, S_3 and S_4, by using the software package *GOstats* (Falcon and Gentleman, 2007). As suggested in (Gentleman, 2004), we did not apply multiple-hypotheses correction, but we set the significance at a quite stringent level ($\alpha = 0.001$) to improve sensitivity and to reduce the number of false positives. Then, for

each clustering solution $S_i, 1 \leq i \leq 4$, we merged the GO terms that we found significantly overrepresented in each cluster. Lastly we compared the set of GO terms that biologically characterize each clustering solution S_i.

	S_1	S_2	S_3	S_4
S_1	-	0	5	23
S_2	0	-	5	23
S_3	20	20	-	20
S_4	39	39	20	-

Table 2.2: Comparison of the number of GO BP terms differentially overrepresented at 0.001 significance level between the different clustering solutions S_i.

We found that 233, 233, 248 and 249 GO terms were significantly overrepresented in the four unique solutions. These turn out to be quite similar, but with some relevant differences. Although no difference is registered between clustering S_1 and S_2, we did find significant differences between all the other clustering solutions: for instance, 5 terms are overrepresented in S_1 but not in S_3, and 20 GO BP terms are overrepresented in S_3 but not in S_1 (Table 2.3.3). Clusterings S_1 and S_4 differ for 23 terms overrepresented in S_1 but not overrepresented in S_4 and 39 GO BP terms in the opposite direction. In other words, considering this last comparison, this means that about $1/4$ of the GO terms enriched in the two different clustering solutions are different between clusterings S_1 and S_4. As an example, looking at the terms overrepresented in clustering S_4 but not overrepresented in S_1 (Table 2.3.3), we see that these functional classes are characterized by biological processes involved in the structural organization of cellular components and by its related anabolic/catabolic processes. Other biologically significant differences can be detected by comparing the other clustering solutions (data not shown). In other words, different HC solutions do lead to different biological characterization of clustering results, highlighting the relevance of the non-uniqueness of the solution in the bioinformatics domain.

2.3.4 Discussion

As shown in Section 2.3.3, the non-uniqueness of the solution is a critical problem, since it can make results inconsistent, leading to different interpretations of the same data depending on the order in which the data are presented. To avoid this, *all* the possible dendrograms that result from different MD pairs could be considered, but this is not a feasible approach. In fact, in the worst case, we obtain $p = N/2$ non-critical pairs at the first clustering step, from which $(N/2)!$ dendrograms are generated, leading to a complexity of $O(N!)$.

The method introduced here allows to greatly reduce the number of generated dendrograms, without sacrificing completeness. This is achieved by a careful analysis of the ties

GO identifier	Definition
GO:0016043	cellular component organization
GO:0006996	organelle organization
GO:0022411	cellular component disassembly
GO:0032984	macromolecular complex disassembly
GO:0034623	cellular macromolecular complex disassembly
GO:0006270	DNA replication initiation
GO:0000082	G1/S transition of mitotic cell cycle
GO:0010926	anatomical structure formation
GO:0022607	cellular component assembly
GO:0000128	flocculation
GO:0000501	flocculation via molecular cell wall interaction
GO:0006268	DNA unwinding during replication
GO:0006873	cellular ion homeostasis
GO:0016584	nucleosome positioning
GO:0019725	cellular homeostasis
GO:0032392	DNA geometric change
GO:0032508	DNA duplex unwinding
GO:0032784	regulation of RNA elongation
GO:0032786	positive regulation of RNA elongation
GO:0055082	cellular chemical homeostasis
GO:0031124	mRNA 3'-end processing
GO:0007154	cell communication
GO:0007165	signal transduction
GO:0007047	cell wall organization
GO:0045229	external encapsulating structure organization
GO:0009056	catabolic process
GO:0006970	response to osmotic stress
GO:0006520	cellular amino acid metabolic process
GO:0044106	cellular amine metabolic process
GO:0006413	translational initiation
GO:0006417	regulation of translation
GO:0010608	post-transcriptional regulation of gene expression
GO:0032268	regulation of cellular protein metabolic process
GO:0051246	regulation of protein metabolic process
GO:0000463	maturation of LSU-rRNA from tric. rRNA transcript
GO:0000470	maturation of LSU-rRNA
GO:0006913	nucleocytoplasmic transport
GO:0051169	nuclear transport
GO:0001522	pseudouridine synthesis

Table 2.3: GO terms overrepresented in S_4 but not in S_1.

that arise in the clustering process. More precisely, we showed that it is possible to identify the equivalence classes over dendrograms, according to Def. 2.3.4, and to generate a single dendrogram for each class. The reduction in the number of dendrograms is relevant: only 128 dendrograms were generated for the dataset of Section 2.3.3 (and 98 for that of Section 2.3.3). On the contrary, when considering all MD pairs, or even the critical pairs only (equivalent and non-equivalent ones), the clustering procedure stopped when 100,000 dendrograms were generated, because of memory saturation.

This reduction has been obtained by limiting the number of dendrograms that must be

fully developed, although for each new dendrogram all the data that identify the clustering state have to be saved. The dominant cost is represented by the dissimilarity matrix, which is $O(N^2)$, at least at the first clustering steps. Overall, the algorithm has therefore a complexity of $O(qN^2)$, where q is the number of non-equivalent pairs encountered along the clustering process (and of the generated dendrograms); we explicitly remark that usually $q << p$. This figure is much smaller than $O(N!)$, obtained when developing all the dendrograms stemming from MD pairs. We explicitly notice that, after all clustering solutions have been generated, an additional step is required to identify the *unique* solutions: in fact, some solutions, even though they derive from different dendrograms, may be constituted of the same clusters.

We remark here that equivalent dendrograms are not *identical* dendrograms: by choosing one representative for each equivalence class, we do compress information. Let us consider, for instance, Figs. 2.5a and 2.5b, that show two equivalent dendrograms. If a 2-cluster solution is required, different solutions would be obtained from those dendrograms. Although in both cases the two clusters will be merged into the same cluster in the subsequent clustering steps, these two equivalent clusterings do exist in the intermediate steps. Notice that this is true even for identical dendrograms: the dendrogram shown in Fig. 2.4a could have been obtained by either merging the yellow elements first, or the purple ones. According to which pair was selected first a different 3-cluster solution is obtained. In these cases, equivalent pairs should be tracked at each clustering step. If a dendrogram has to be cut at a level for which there exist equivalent pairs that have not converged into the same cluster yet, an additional "backtracking" step would be required for retrieving the equivalent clusterings associated with that cut.

Also notice that, at each step, we identify all the MD pairs, but only one pair of clusters is merged. An additional speed-up could be attained if we merged non-critical pairs, and all the objects belonging to the same equivalence class, in one step. This can be seen as collapsing multiple clustering steps into one. However, this would not allow for an easy backtracking (see above), and deviates from the traditional HC paradigm based on iterative merging of the *two* closest objects, and therefore it has not been implemented.

Our algorithm is described here with Ward's dissimilarity measure but it can be applied to other measures as well (see Section A.2 in the Appendix), as long as they are not prone to inversion (Morgan and Ray, 1995). This occurs when the sequence of merging coefficients is non-monotonic: in this case, the fact that Eqs. 2.24a and 2.24b hold at the current step does not guarantee equivalence, as a subsequent merging operation could produce a cluster C_z that is closer to $\{C_i, C_j\}$ than C_k is. The monotonicity requirement rules out centroid and median linkage clustering, whereas simple, complete, group and weighted group average linkages can be successfully employed with the presented method.

The proposed approach could also be extended to the case of real-valued data sets. Although in this case it is unlikely that exact ties occur, it is possible that the data are affected by noise. We may therefore introduce a tolerance value and consider two distance values to be equal when they differ less than ε (ε depending on the noise characteristics). The rationale

for this choice is that the correct clustering solution should not be dictated by noise.

It should be noticed that the problem of non-uniqueness of the solution can affect central clustering, too. When using a trial-based (on-line) approach, the presentation order of the data points is relevant as each data point moves the cluster centroids, thus influencing every subsequent step. In batch approaches, on the other side, the existence of ties between a data point and two, or more, cluster centroids can lead to multiple solutions also in this case, as a ranking of the point with respect to the clusters is arbitrarily chosen. An approach similar to the one described here should be developed for central clustering too, in order to keep track of these possible alternative solutions.

2.4 Further methodological work on meta-analysis

In the previous section we have described a modification of classical HC developed to address the problem of non-uniqueness of the solution; this provides the meta-analytic procedure with a stable clustering result that is then subject to further analyses, e.g. statistical testing, for establishing the functional role of each cluster (and corresponding brain region).

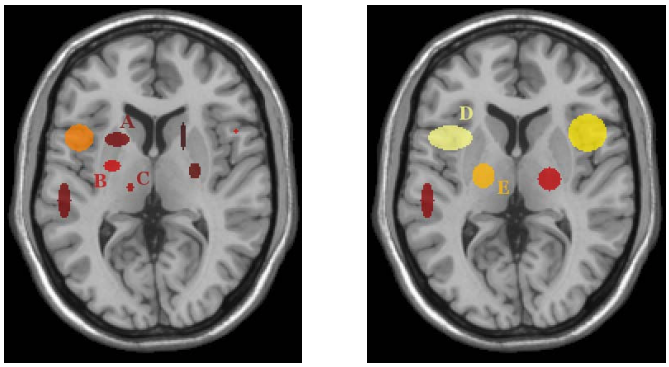
During our work on these topics, we also developed some additional tools to support the execution of a meta-analysis of functional neuroimaging data, namely MATLAB scripts for visualizing clusters as blobs superimposed on the cerebral volume, and for automatically extracting an anatomical label for each of them. Each blob is represented as an ellipsoid centered on the mean coordinate of its cluster, and whose semi-axes correspond to the standard deviation of the cluster in the three directions; depending on the intended use of the figure, a color code is used to either represent the cardinality of the cluster (as in Fig. 2.7), or to distinguish among clusters belonging to different categories (see, for instance, Fig. 3.4 in the next chapter). The "blob image" can be overlaid on a structural image of the brain using, for instance, MRICro (Rorden and Brett, 2000). The anatomical atlases available in MRICro (AAL and Brodmann templates) were employed to derive anatomical labels: such atlases are polled with the mean coordinate of each cluster to obtain the codes (both the anatomical designation, and Brodmann number) corresponding to the brain region to which that coordinate belongs.

Aside from these minor, although practically useful, contributions, we also reasoned on a further improvement of the meta-analytic process. The clustering step is, up to now, purely spatial: that is, no other information is given to the clustering algorithm, apart from the position of each activation peak on the cerebral volume. Clusters are formed exclusively on the basis of the spatial proximity of such points. At first, this may appear as a very advantageous feature, as no additional information encoded by the experimenter and, as such, potentially biased toward a particular hypothesis, can affect the clustering process, which remains therefore totally objective. However, the lack of any additional information can result in the production of clusters that are *anatomically implausible*: that is, clusters that collect activation peaks belonging to different anatomical areas. The reason for this is that the clustering process is unaware of any anatomical subdivision existing in the brain: it only operates based on Eu-

clidean distances, and cannot know where an anatomical boundary exists that should not be crossed. In most cases, cluster "crossings" constitute a mild, or no, anatomical violation, such as when a cluster groups points belonging to two gyri of the same lobe. Potentially critical, but generally still acceptable, are clusters that span an inter-lobe boundary, such as a cluster situated in an occipito-temporal area. However, in some cases the anatomical plausibility of a cluster might be strongly disputed, if it includes activation coordinates belonging to highly incompatible brain regions, such as the occipital lobe and the cerebellum. Conclusions on the functional role of such a cluster could be problematic, as it most likely represents the fusion of two different neural populations, rather than a functionally homogeneous circuit.

Thus, it appears that the introduction of some anatomical knowledge into the clustering process would be profitable to the development of a sound meta-analytic approach in the field of functional neuroimaging. We started investigating how anatomical constraints can be incorporated into clustering in (Radaelli, 2010). A simple solution was adopted there, based on a user-defined pre-partitioning of the input data set. The experimenter, based on general anatomical considerations and specific requirements of the analysis he is conducting, may deem some brain regions to be strongly incompatible with one another. In other words, he might want to distinguish among groups of areas, defined such that clusters including peaks from different groups are explicitly ruled out; let us refer to these groups as AGs (for anatomical groupings). Anatomical constraints are therefore expressed in the form of mutually exclusive AGs of anatomical areas. For instance, two AGs, one including all areas in the left hemisphere, and one with the areas in the right hemisphere, might be indicated if the analysis has a particular focus on lateralization aspects; or, an AG containing the cerebellum only, with the rest of the brain in a different group might be created if the only constraint is not to cross the cerebellum-forebrain boundary. The user is thus allowed to define how many AGs as he thinks are necessary; this results in a partitioning of the original input dataset into separate datasets, each corresponding to an user-defined AG. Based on the AG configuration file, an automatic procedure (based on the automatic labelling script described above) scans the input dataset and determines the anatomical location of each activation peak, and therefore the AG it belongs to; a different input file is then created for all peaks falling into an AG. This pre-processing step is then followed by the clustering step, which operates on each input file separately; lastly, the clusters produced by each clustering process are integrated in the final result (as each clustering process denotes its clusters with a progressive ID, starting from 1, renomination of clusters with unique IDs is necessary). It should be noticed that anatomical templates do not cover the whole brain: there are regions having no anatomical label, and these can represent a problem for the dataset partitioning procedure. For unlabelled peaks, it is therefore required that the user manually assigns each of them to one of the defined AGs.

Figure 2.8 shows a comparison of the clustering results obtained with our clustering algorithm when five AGs are defined (Fig. 2.8(a)), two including the left and the right thalamus, respectively, two including the basal ganglia for each hemisphere, and one for the remaining regions, and when no anatomical constraints are introduced (Fig. 2.8(b)). Here, the employed



(a) (b)

Figure 2.8: Comparison of clustering results in a 494 peaks dataset, when introducing anatomical constraints – Panel (a) – and with unconstrained clustering – Panel (b). In the latter case, two clusters (D and E) spanning over incompatible brain areas, as dictated by the user, are obtained. The introduction of anatomical constraints, on the other hand, effectively produces anatomically plausible clusters (A, B, and C).

dataset was constituted of 494 activation coordinates coming from a preliminary dataset for our meta-analysis on motor imaging (Invernizzi et al., In Preparation). As it can be seen, if no anatomical constraints are introduced, we do get "implausible" clusters (where implausibility is here intended as a violation of the set constraints): looking at the left hemisphere, we can identify one cluster (denoted with letter D) that contains both peaks from the basal ganglia and the cortex, and one (cluster E) that includes both points in the thalamus and basal ganglia. Anatomical constraints are effective in preventing the creation of such inter-boundary clusters, and in this example result in the identification of two clusters (A and B) in the left basal ganglia, and one (C) in the left thalamus.

The above illustrated solution, based on pre-partitioning the input dataset, is simple and yet of practical utility for a meta-analysis user, as it allows anatomical constraints to be naturally declared and introduced into the clustering process, so as to obtain clusters that do not cross anatomical boundaries that the experimenter regards to be critical for a correct interpretation of the results. With no such mechanism, it might be required that the analyst manually splits anatomically implausible clusters (this was done in the study reported in Chapter 3); an automatic procedure, that also allows AG configuration files to be saved and later re-used as templates, helps simplify the operation, and reduce the time required to perform the meta-analysis. This preliminary work could be extended to allow for more graded constraints to be introduced. We have already mentioned that, aside from regions that, with high level of agreement, can be considered strongly incompatible from a functional-anatomical perspective, there are regions that we might want to keep separated – but these boundaries are not so critical and might be crossed on occasion. A pre-partitioning approach clearly cannot accommodate these cases. Ideally, it should be desirable to define a hybrid notion of dissimilarity

function to be used in clustering, which incorporates both purely spatial information, and "semantic" one. Here semantic information is used to refer to any kind of knowledge specific of the application domain at hand that could be effectively exploited by the clustering process: in our case, semantic knowledge coincides with knowledge about the anatomical organization of the cerebral volume. The two notions of spatial distance and semantic dissimilarity would then combine to determine the overall affinity of two points in the brain. However, determining how such combined dissimilarity measure should be defined is not straightforward: exploratory simulations conducted on a modification of the fuzzy c-means algorithm (see Section 2.2.3), where "anatomical weights" that multiply spatial distances were introduced (Radaelli, 2010), revealed that finding a good balance between the two sources of information is generally difficult, especially if initialization of centroids is not optimal. However, no extensive experimentations on this topic have been carried out yet. Alternative approaches that could be investigated include the introduction of an additive (rather than multiplicative) semantic contribution to the combined dissimilarity measure, and the use of topographic distances on the brain in place of Euclidean ones, to take into account the three-dimensional geometry, including sulci and gyri, of the cerebral volume, together with a measure of the degree of connectivity between regions.

2.5 Conclusion

In this chapter we have presented our methodological work on functional neuroimaging meta-analysis. The main result is the development of a novel clustering algorithm that solves the problem of non-uniqueness of the solution in HC, due to the potential presence of ties in the dissimilarity matrix. We have shown that by defining an adequate equivalence relation over the dendrograms stemming from the data, all the significantly different clusterings can be generated with polynomial complexity. This allows obtaining a unique solution independently of the data presentation order, which guarantees a unique interpretation of the data. As illustrated by our experimental results, this is of particular value in neuroimaging and bioinformatics, but it can also be applied to other domains to which HC is suitable.

We have also described some preliminary work aimed at improving the meta-analytic process from the point of view of the anatomical plausibility of resulting clusters. A classical clustering approach takes into account only spatial distances in an abstract space, and therefore merges all those data points that have small distance value, regardless of any additional constraint that might exist in the particular application domain. In our case, these constraints are anatomical, meaning that some boundaries between brain regions should not be crossed when forming clusters: if this happens, an anatomical heterogeneous cluster results, and conclusions that might be derived on its functional role could be misleading. We have described a basic pre-processing step that takes anatomical constraints from the user and creates, based on them, a partitioning of the input dataset, so that the clustering procedure can be run on each subset of data separately. This constitute a first, effective way to ensure that user-defined

anatomical constraints on clusters are not violated. We have also discussed some additional directions that might be taken in exploring the topic of semantic clustering for functional neuroimaging meta-analysis.

The whole meta-analytic method would also benefit from careful code optimization for the clustering routine, to reduce execution times and memory occupation, and from the development of a complete, integrated meta-analysis toolbox that could support neuroscientists in their investigations by providing an automated procedure for performing the set of operations involved by such analyses. In perspective, we believe that the work described here constitute a meaningful contribution to the improvement of the methodology employed in clustering-based meta-analysis of functional neuroimaging data – a powerful tool for aggregating and advancing knowledge about the neural implementation of human cognitive faculties.

Chapter 3

Reading the reading brain: a new meta-analysis of functional imaging data on reading*

"Eddie doubted if the gunslinger could actually read much of the document; this world's written words would always be mostly mystery to him"
— From "The Dark Tower", by Stephen King, 1947–

3.1 Introduction

The processes involved in single word reading, i.e. in the mapping of an orthographic representation to its phonological form, have been studied extensively, both with behavioural, modelling, and neuroimaging experiments.

Lesion studies have provided a neurological taxonomy of acquired reading disorders of the classical syndromes such as pure alexia, dyslexia and agraphia, etc. (see Cappa and Vignolo, 1999, for a review); empirical observations in brain damaged patients (reviews in Coltheart, 1982; Denes et al., 1999) have constituted the primary basis for the development of cognitive models of reading among which the highly influential dual-route models, which predicate that letter strings are decoded either through an orthographic lexical routine or via a sub-lexical procedure (see e.g. Baron and Strawson, 1976; Besner and Smith, 1992; Coltheart, 1978, 1985; Forster, 1976; Morton and Patterson, 1980; Paap and Noel, 1991; Patterson and Morton, 1985a). This model has also been implemented in a computational format, the well-known Dual-Route Cascaded (DRC) model (Coltheart et al., 1993, 2001), supporting the validity of the framework. However, the implementation of the model appears somewhat artificial[1]: a connectionist network serving as a lexical database coupled with a serial mech-

*This chapter contains an early version of a manuscript currently under revision for resubmission (Cattinelli et al., Under Revision–a); the accompanying Supplementary Materials can be found in Appendix A.3. Note that more details on the psychology of reading (such as the definition of pseudoword, or extensive treatment of the major models of reading) are given in Chapter 5. The employed clustering method is detailed in Chapter 2.

[1]We refer in particular to the fact that DRC combines two different computational paradigms (sequential rule-based processing, and parallel processing in a neural network) for realizing the two routes.

anism of symbolic rules acting as the sub-lexical route, and a quite large number of free parameters (see discussion in Zorzi, 2005). Moreover, no learning process is implemented: both the lexical entries and the GPC (grapheme-phoneme correspondence) rules are hand-coded.

Comparably good performance is produced by an alternative model, the Triangle Model (Plaut et al., 1996; Seidenberg and McClelland, 1989), which postulates a common process for managing every kind of orthographic string. Even though the most recent implementation of the model (Plaut et al., 1996), the so-called PMSP model (from the initials of the authors) improves its predecessor (Seidenberg and McClelland, 1989), especially on pseudoword reading, the model is still unable to correctly simulate serial effects, such as the position of irregularity effect, whereby words having an irregular grapheme-phoneme correspondence in earlier positions are read with longer latencies than words for which the irregularity appears later in the string (Coltheart and Rastle, 1994; Cortese, 1998; Rastle and Coltheart, 1999). These are just the two best known among such models: more recent neural network models tried to overcome some of the above mentioned limitations (e.g. Perry et al., 2007; Zorzi et al., 1998) – (for a review, see Zorzi, 2005). It is worth noting that these models were developed with a special attention to the English orthography, which has features that make it unique, and that only monosyllabic words are currently handled by these implementations (although very recent work has targeted bysillabic words too (Perry et al., 2010)). Definitive adjudication among cognitive models of reading has not been achieved yet; moreover, the links between the proposed cognitive modules and their possible anatomo-physiological correlates have been established, at best, only loosely.

In fact, lesion studies have not been very informative as yet in supporting fine grained cognitive neuropsychology theories of reading. Another interesting source of information, because of the much more detailed spatial resolution and the physiological nature of its experiments, is functional imaging, which has the potentials of providing a better anatomical testbed for cognitive theories of reading. There have been now 20 years of imaging experiments on this topic, with more than 40 studies on alphabetic orthographies alone. Interestingly, none of the published studies can claim conclusively to have demonstrated the validity of one or the other cognitive/computational model. Indeed, the thrust of this enormous experimental work and the progress achieved so far in the understanding of the physiology of reading has not been assessed fully as yet, although the main neuroanatomic components of the reading network are known.

Input processing is most likely carried out by the ventral stream of the visual system of the left hemisphere. In recent years, special attention has been given to the posterior part of the left midfusiform gyrus as specifically responsible for word recognition, hence the name of Visual Word Form Area (Cohen et al., 2000, 2002). However, subsequent studies have shown that this area is not exclusively activated when reading words, but also in non-orthographic tasks such as picture naming or auditory word repetition (for a review, see Price and Devlin, 2003). The left posterior temporo-parietal cortex including the angular gyrus, as well as the middle and inferior temporal gyri, have been linked to semantic access, but the left angular

gyrus has been also associated with word form recognition (Price, 2000). The left frontal inferior gyrus, crucial in speech production, plays also a role in reading output, together perhaps with the supplementary motor cortex and the cerebellum (Fiez and Petersen, 1998). However, these functional characterizations fall short of providing a detailed neurophysiological account for cognitive models of reading.

With a large dataset such as the one on reading, the question arises as to how to further develop this area of research. One obvious approach is to perform new well-designed experiments to explore specific features in the physiology of reading. For example, one such study is the recent one by Graves and colleagues (2010) reviewed in the Discussion.

On the other hand, one can reconsider the published data to assess which set of results were strong enough to be replicated and to represent genuine new knowledge in the area; indeed a reassessment of the data may in principle bring about new evidence. To this end, a number of excellent review articles have been published (Fiez, 1997; Fiez and Petersen, 1998; Petersen and Fiez, 1993; Price, 2000; Price and Devlin, 2003; Price and Mechelli, 2005; Pugh et al., 2000). However, these usually lack a formal assessment on the replicability of the previously published data, leaving a good deal of uncertainty when one faces contradictory findings for very similar experimental protocols and techniques.

One possible alternative to qualitative reviews is represented by formal meta-analyses: with imaging datasets of the size of that on reading, formal meta-analyses have the potential of "separating the wheat from the chaff" in distinguishing solid observations from more ephemeral ones, and maybe to permit new insights on the processes subserved by a given brain region.

In brain imaging, meta-analyses are generally used to identify groups of activation peaks that fall sufficiently close in stereotactic space (given the spatial resolution of the dataset) to be interpreted as reflecting a common functional-anatomical entity. Once identified by a clustering process, the functional significance of any given region needs to be scrutinized on the basis of the background information about the experiments that generated the activation peaks belonging to the cluster. This, as we will discuss later, is a crucial point of the process.

To date, there is only one published meta-analysis on reading, based on hierarchical clustering, which aimed at correlating imaging data with a (neuro)psychological model (Jobard et al., 2003): they assessed 35 neuroimaging studies in order to evaluate whether fMRI and PET data could altogether support the dual-route theory of reading. A mixture of orthographies were considered, primarily alphabetic irregular (English), but also non-alphabetic orthographies such as Kanji.

A total of 622 activation coordinates was collected and automatically grouped, using a hierarchical clustering procedure (Ward, 1963), in 55 clusters; of these, only 11 clusters, corresponding to areas *previously involved*, as they say, in reading, were discussed in the paper. Based on their composition, each of these clusters was assigned to one of the considered reading processes (sub-lexical process, semantics, pre-lexical analysis, and early visual processing). On the basis of these attributions the conclusion was drawn that a dual-route account of

reading is supported by the meta-analysis, suggesting the existence of two distinct networks in the left hemisphere: one subserving a sub-lexical grapheme-to-phoneme routine (posterior and mid superior temporal gyrus, mid part of the middle temporal gyrus, supramarginal gyrus, and pars opercularis of the inferior frontal gyrus); the other being related to semantic access (inferior temporal gyrus, posterior middle temporal gyrus, and pars triangularis of the inferior frontal gyrus).

While representing a pioneering study, (Jobard et al., 2003) paper has a number of limitations that invited us to reconsider the issue of a meta-analysis on reading. These limitations are discussed in the Supplementary Materials and summarized as follows: first, the paper included data from non-alphabetic orthographies that may not be relevant for a dual-route process of reading; second, the assignment of a cluster to a given process was made only qualitatively, i.e. without any statistical testing; third, there was an a priori decision on which clusters to consider as relevant for the reading process; fourth, the hierarchical clustering algorithm adopted by Jobard et al. (2003) can return different solutions when the input data order is changed (see, for instance, Morgan and Ray, 1995). In addition, a number of further technical issues precluded an exact replication of Jobard's results.

The aim of our study was therefore to re-assess the corpus of imaging data on reading with a new meta-analysis, and try to address as much as possible some of the limitations discussed above.

It is important to notice that the results of a neuroimaging study on reading can often lead to different implications according to how the data are interpreted, especially depending on the specific theoretical assumptions motivating the investigator.

Traditional dual-route theories postulate the existence of two separate mechanisms dedicated to the processing of different categories of orthographic stimuli, calling for the existence of two separate anatomical systems as well. If translated into functional imaging procedures, specific statistical comparisons of the form *exception words* $>$ *pseudowords* should massively activate one neural circuit while the other one should remain relatively silent; a canonical view on the operations of the dual-route model also postulates that the opposite pattern should be observed for the reversed contrast[2]. In a single-mechanism framework, on the contrary, no dramatic segregation of activation patterns would emerge, since the whole reading system would be responsible, to a certain degree, for the correct processing of all categories of stimuli; at the same time, we could expect a greater engagement of areas involved in semantic processing for words (both regular and exception) compared to pseudowords.

However, these are just preliminary and somewhat naive observations on which kind of activation patterns one should expect when assuming the validity of a certain theory of reading. First of all, orthographic stimuli do not differ only on the basis of their lexicality, or degree of regularity. A large number of psycholinguistic variables are involved: frequency, length, imageability, emotional content, orthographic/phonological neighbourhood size, and possibly many others (Forster and Chambers, 1973; Frederiksen and Kroll, 1976; Glushko,

[2]The assumption that a pseudoword is not processed at all by a lexical route has been questioned by the defenders of the reading-by-analogy model (Kay and Marcel, 1981).

1979; Landis, 2006; Strain et al., 1995; Weekes, 1997; Yates, 2005).

The same concept of lexicality may be misleading: we could regard words as "common" stimuli that our reading system recognizes as familiar, whereas pseudowords are perceived as unfamiliar stimuli that require special attention - and this would qualify them as especially difficult items to handle. From this perspective, lexicality might eventually reduce to a binarization of the concept of frequency (of exposure). Regularity itself is linked to frequency as well: a regular grapheme-to-phoneme correspondence is one that occurs frequently in a given orthography, and therefore an "easy" one. In brief, confounding variables can make it difficult to give a unique interpretation of observed patterns of activation: what could at first appear as a specificity for lexicality and/or regularity might as well be an effect of frequency, or of other features of the set of stimuli in hand.

Moreover, most published neuroimaging studies rest on the fact that a greater activation in a given brain area is a sign of an exclusive, or specific, involvement of that area in the processing of a particular kind of stimulus, or task[3].

However, another hypothesis could be entertained. An area might be involved, say, in the reading of all kinds of orthographic stimuli; yet, when presented with a difficult item (as we might consider a pseudoword to be, by comparison with a well-known word), this area would reasonably require a larger amount of neural energy in order to correctly process it - more neural activity translating into larger BOLD signal. On the basis of the measured signal, one could conclude either that this area is particularly responsive to pseudowords, or that it is part of a generic reading network which shows a modulation due to item difficulty. The first conclusion would be consistent with a dual-route theory, and would identify that particular brain area as part of the circuit responsible for the application of grapheme-to-phoneme correspondence rules. The second interpretation, on the other hand, would rather support a single-route account, as no arbitrary segregation according to lexicality status is advanced.

Figure 3.1 provides a hypothetical account of the processes involved in reading various kinds of words in a dual-route perspective at a physiological level, which is different from a lesion-based account in many respects.

A word with regular spelling can be correctly processed by both a putative sub-lexical and a lexical route. Activation of the semantic system has also to be considered. As no conflict exists at the level of phonological output between the results of the operations of the two components, only a mild activation of the phonological output can be assumed.

For irregular words, the situation is similar (the system has no a priori information about the nature of the stimuli – regular vs. irregular, word vs. pseudoword –, so all input strings

[3]The idea that imaging studies should always reveal greater regional activity when the stimulus of a given category is exposed to groups of neurons with "experience" for that category is a naive one for many reasons: the brain is not all made by "grand-mother"-like cells; local circuitry may be organized hierarchically, holding representations of different size; high-level aspects of cognitive processes may emerge from distributed activity whose variations may be relevant in the temporal domain rather than in a topographically localized domain. Yet, the evidence from neuropsychological patients with focal lesions invites also to consider the local aspect of neuronal computation in high-order cognitive functions.

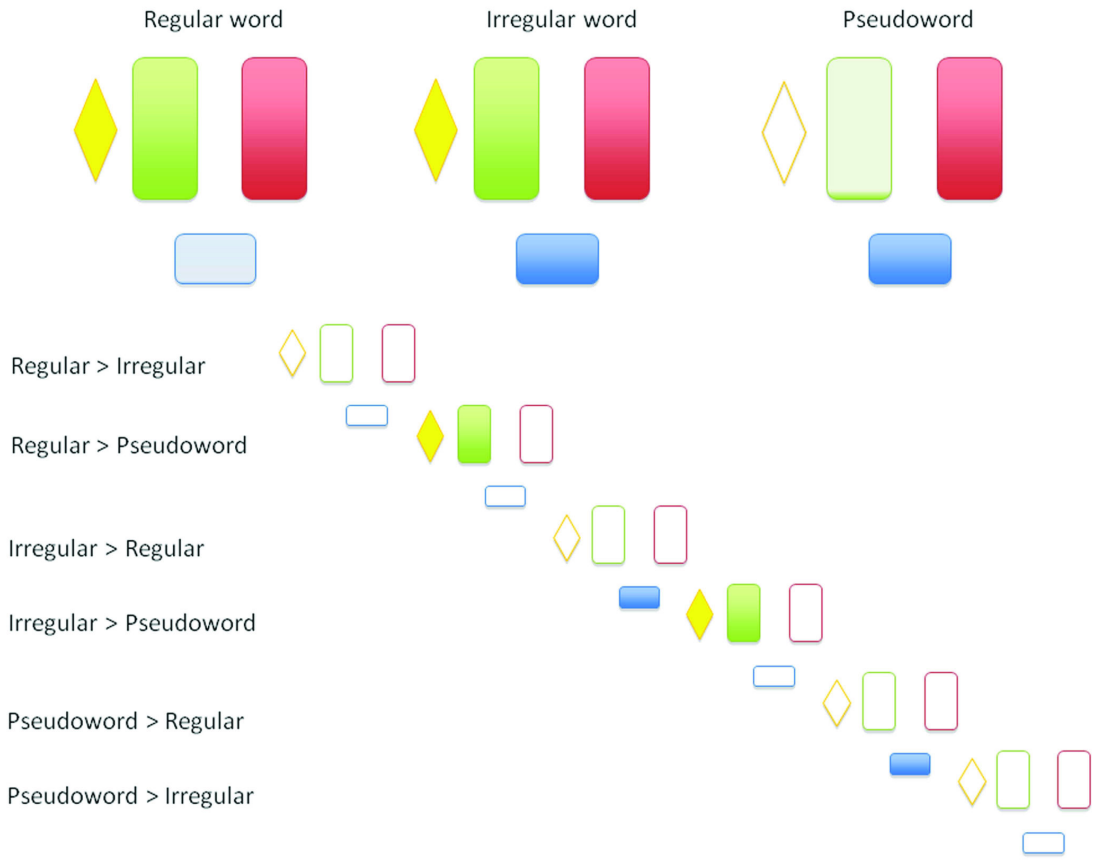


Figure 3.1: Some predictions based on a traditional dual-route model. The green box represents the lexical route, linking orthographic whole-word forms to the corresponding phonological forms; the red box represents the grapheme-to-phoneme conversion pathway; the blue box represents the phoneme output system; and the yellow diamond represents the semantic system. The predicted relative involvement of these systems when comparing different categories of written stimuli is shown in the lower part of the figure.

must be processed by both routes), the only difference being in the fact that the phonological output of the sub-lexical route is incorrect, causing a competition with the output of a lexical route. This competition would be maximized for low-frequency irregular words for which the lexical route, due to its sensitivity to word frequency, would have longer computation times. Conflicting phonological outputs may translate into a stronger activation of the phonological output system, and stronger BOLD signal.

Finally, pseudoword reading would rely on the sub-lexical routine with a marginal involvement of the lexical route depending on the partial activation of similar real words. This partial activation may still have an impact on the level of activity in phonological output processing due to conflict, as for the irregular words. Figure 3.1 offers a set of possible scenarios derived from the comparisons of the patterns of activation for the different kinds of stimuli.

It can be seen that there is not an opportunity, through a simple series of direct comparisons of brain activations, to identify a distinction between a lexical from a sub-lexical route,

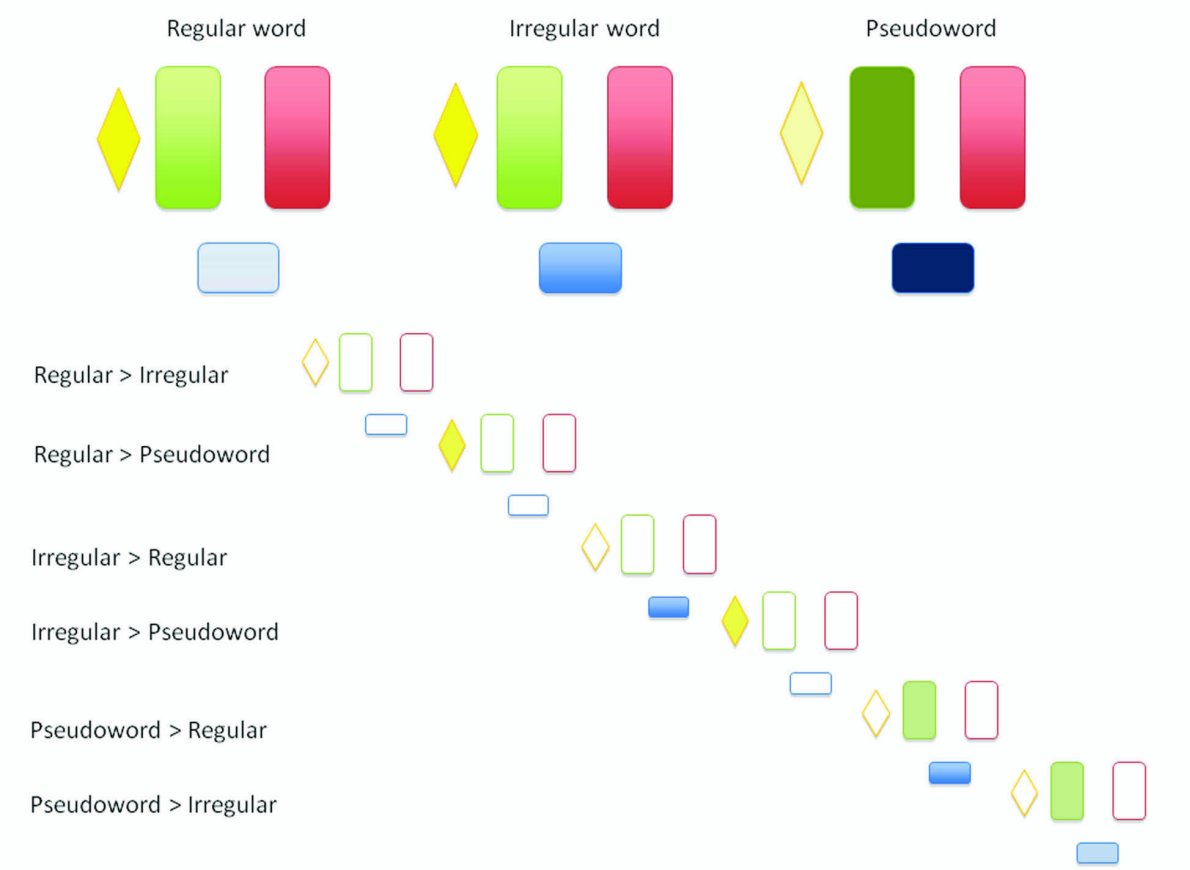


Figure 3.2: Some predictions based on a modified version of the classical dual-route model. Components of the model are depicted as explained in the caption of Fig. 3.1. In this account, pseudoword reading activates the lexical system much more than word reading, since an extensive search into the lexicon is performed in order to find the (non-existent) entry corresponding to the stimulus.

nor the possibility of drawing a distinction between a semantic and a lexical component in word reading. In fact, extra knowledge is probably needed about the nature of certain brain areas to infer that they hold a lexical representation deprived of semantic information.

From the figure it can be also seen that one can envisage a common effect of a larger activation of output phonology for both irregular words (compared with regular) and pseudowords compared with regular words: this shared effect could be more generically labelled as an "item difficulty" effect.

An account of the predictions of a modified dual-route model is given in Figure 3.2. Here it is assumed that pseudoword reading also capitalizes on a by-analogy strategy (Kay and Marcel, 1981) whereby an orthographic lexicon is extensively interrogated on the basis of incomplete correspondences of graphemes or larger sub-word units (partial matching between strings). The extensive search through orthographic lexical representation would lead to a counterintuitive hyperactivation of the ensuing neural substrates. This was the proposal made by Paulesu et al. (2000) on the basis of PET activation data in readers from different cul-

tures, the English and the Italian one. Paradoxically, perhaps, this modified model predicts the possibility of identifying the neural counterparts for orthographic lexical representations better than the original dual-route model.

On the other hand, a connectionist model like the PMSP model (Plaut et al., 1996) predicts a more distributed involvement of the reading system in the processing of all kinds of orthographic strings, based on the mutual interaction of regions coding for different sources of information (orthographic, phonological, semantic). Words would differ from pseudowords not because they can be retrieved in an orthographic lexicon collecting all known whole-word representations, but because they can count also on contributions from the semantic system to be correctly read.

None of these hypotheses, however, are uncontroversial; here they were discussed in order to illustrate the problems that one faces when trying to provide a demonstration of the neural implementations of popular cognitive models of reading.

In this meta-analysis, we took the above considerations into account in order to describe the most solid results in terms of functional anatomy of reading and to assess the validity of the competing theories: in particular, we focused not only on (apparent) lexicality and regularity effects, but we also looked for difficulty effects across the reading network, which may support a different interpretation of the reading processes than the traditional dual-route one.

As the reader shall see, our results show that the functional neuroimaging data examined in the meta-analysis do support the view that reading occurs through a distributed system more compatible with the assumptions of the PMSP model rather than through a system showing the degree of modularity implied by the classical dual-route model.

3.2 Materials and methods

3.2.1 Data collection

This meta-analysis considers the neuroimaging literature on reading starting from the earliest pioneering studies (late '80s) up to 2008. Neuroimaging experiments present a large amount of variability as far as the imaging methodology, experimental design, nature of the stimuli, stimulation procedures, task demands, which may make the resulting activation data hardly comparable. We therefore determined a list of inclusion criteria which had to be met for a study, or a specific result derived from a given statistical comparison, to be included in this meta-analysis:

- *Subjects*: we considered only studies including at least six normal adult subjects; no data from neuropsychological patients were used.
- *Scanning technique*: fMRI or PET.
- *Stimuli*: single words only (e.g., no word pairs, or sentences).

- Alphabetic orthographies only were considered.

- *Data analysis*: only whole-brain analyses were considered (i.e., no ROI approaches). P-values were also checked: for simple comparisons (i.e., task $>$ baseline), activation data were included in our database only for P-values at least < 0.001 (uncorrected for multiple comparisons); for direct comparisons (e. g., words $>$ pseudowords), we accepted a $p < 0.01$ (uncorrected) threshold. In addition, we considered only univariate analyses.

- *Tasks*: only four classes of tasks were considered: reading (aloud or silently, explicitly or implicitly), lexical decision, phonological decision, and semantic tasks. Other tasks such as priming, and other tasks involving the processing or pairs or triplets of stimuli, were not considered, as we focused on single word processing only.

- *Contrasts*: To maximize the relevance and specificity of the data for the identification of specific processes involved in reading, only the results of certain kinds of statistical comparisons were considered: for example, we excluded conjunctions (e.g. Reading and Lexical Decision $>$ Baseline), comparisons among tasks rather than classes of stimuli (e.g. Reading $>$ Lexical Decision), and main effects (e.g. Reading words and pseudowords $>$ Baseline). As a guiding principle, we gave preference to the most specific comparisons: for instance, when a paper reported activation data from both the contrast main effect of Words (regular & irregular) $>$ Baseline, and the contrast Irregular Words $>$ Baseline or Regular Words $>$ Baseline, only data from the simple effect contrasts were used.

By applying such criteria, we collected a total of 1176 activation peaks, coming from 35 neuroimaging studies. This is the same number of studies as in (Jobard et al., 2003) as a few studies were excluded following the application of the above criteria but, on the other hand, additional and more recent ones were included in our meta-analysis. As a result our final working dataset was made of almost twice the number of activation peaks as in (Jobard et al., 2003), at the same time showing a higher degree of homogeneity. Table 3.1 lists the 35 studies on which this meta-analysis was performed, along with all relevant information.

For each activation peak, we recorded all relevant information about the statistical comparison that generated it. In particular, each peak was classified according to the following variables of interests: lexicality (word or nonword); regularity; frequency (also including orthographic neighbourhood size); syllabic length; type of orthography (deep or shallow). We focused on these specific variables as they are known to modulate naming latencies, and thus arguably the difficulty in decoding an orthographic stimulus (Ferrand and New, 2003; Forster and Chambers, 1973; Glushko, 1979; McCann and Besner, 1987; Paap and Noel, 1991; Paulesu et al., 2000; Seidenberg et al., 1984; Taraban and McClelland, 1987).

Of course, it would have been impossible to take full account of these variables and to evaluate their impact on the data in a formal statistical manner: however, this information proved useful for a qualitative assessment of the possible functional meaning of a cluster.

Reference	*Technique*	*Subjects*	*Design*
(Beauregard et al., 1997)	PET	10 men	Blocked
(Binder et al., 2003)	fMRI	15 women, 9 men	Event-related
(Binder et al., 2005)	fMRI	12 women, 12 men	Event-related
(Bookheimer et al., 1995)	PET	8 men, 8 women	Blocked
(Buchel et al., 1998)	PET	6	Blocked
(Carreiras et al., 2006)	fMRI	10 women, 6 men	Blocked
(Chee et al., 1999)	fMRI	5 men, 3 women (but one subject excluded)	Blocked
(Cohen et al., 2002)	fMRI	6 women, 1 man (first exp.)	Blocked (first exp.)
(Fiebach et al., 2002)	fMRI	5 women, 7 men	Event-related
(Fiebach et al., 2007)	fMRI	8 women, 8 men	Event-related
(Fiez et al., 1999)	PET	5 women, 6 men	Blocked
(Gates and Yoon, 2005)	fMRI	9	Blocked
(Hagoort et al., 1999)	PET	3 women, 8 men	Blocked
(Herbster et al., 1997)	PET	5 men, 5 women	Blocked
(Howard et al., 1992)	PET	7 men, 5 women	Blocked
(Jessen et al., 1999)	fMRI	5 women, 7 men	Event-related
(Joubert et al., 2004)	fMRI	10 men	Blocked
(Kiehl et al., 1999)	fMRI	6 men	Blocked
(Kuchinke et al., 2005)	fMRI	12 women, 8 men	Event-related
(Mechelli et al., 2000)	fMRI	1 woman, 5 men	Blocked
(Mechelli et al., 2003)	fMRI	7, 13 (2 exp. varying stimulus duration or rate)	Blocked
(Mechelli et al., 2005)	fMRI	10 women, 12 men	Blocked
(Menard et al., 1996)	PET	8 men	Blocked
(Meschyan and Hernandez, 2006)	fMRI	7 women, 5 men	Blocked
(Moore and Price, 1999)	PET	8 men	Blocked
(Paulesu et al., 2000)	PET	6 Italian, 6 English	Blocked
(Perani et al., 1999)	PET	14 men	Blocked
(Price et al., 1994)	PET	6 men	Blocked
(Price et al., 1996a)	PET	6 men	Blocked
(Price et al., 1996b)	PET	From 6 to 10	Blocked
(Price et al., 2006)	PET	18 men	Blocked
(Rumsey et al., 1997)	PET	14 men	Blocked
(Valdois et al., 2006)	fMRI	12 (reading), 8 (lexical decision)	Event-related
(Vigneau et al., 2005)	fMRI	23 (simple comparisons), 13 (direct comp.)	Blocked
(Vinckier et al., 2007)	fMRI	8 women, 4 men	Blocked

Table 3.1: Studies whose activation data have been included in the present meta-analysis. For each of them, a bibliographic reference for the paper describing the study, the employed neuroimaging technique, details about recruited subjects, and the type of experimental design are reported.

3.2.2 Template normalization

Most of the latest papers reporting activation data from neuroimaging experiments adopt a stereotactic normalization to the MNI (Montreal Neurological Institute) template. However, some studies, especially the oldest ones, adopted a stereotactic normalization based on the Talairach and Tournoux (1988) space. Therefore, a transformation was needed in order to make activation data from all considered studies comparable.

After determining which template was used in each paper (as explicitly indicated in the

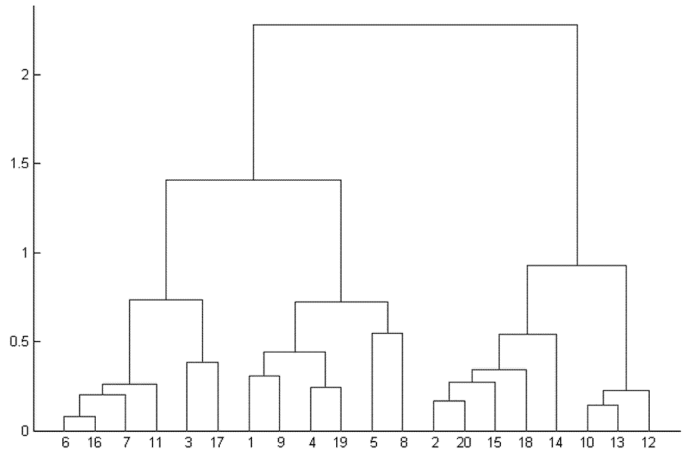


Figure 3.3: Example dendrogram representing the hierarchical clustering process. Values on the x axis are indexes referring to single data points. Lines show which objects have been merged at each clustering step. On the y axis the merging coefficient (dissimilarity value for the pair being merged) is reported.

text, or inferred on the basis of the software used for the statistical analysis[4]), we converted coordinates originally reported in Talairach space to MNI space, by using the *tal2mni* MATLAB script (*http://imaging.mrc-cbu.cam.ac.uk/imaging/MniTalairach*). Our final working dataset, therefore, involved exclusively activation coordinates in MNI space.

3.2.3 Clustering procedure

The 1176 stereotactic coordinates of our database were submitted to a clustering algorithm, in order to automatically group peaks close in space into plausible summary regions of activation. As in (Jobard et al., 2003), we adopted the hierarchical clustering approach (Xu and Wunsch, 2005) with Ward's criterion (Ward, 1963).

In hierarchical clustering, each input data point is initially assigned to a singleton cluster. At each processing step, the two closest (according to the chosen dissimilarity measure) clusters are merged into one. Thus, at each step the number of clusters is decreased by one, until the last step is reached and, eventually, only one cluster, including all input data, is obtained. For each step, the existing clusters represent an admissible grouping of input data: what changes among different steps, is just the number of clusters used to group the data. This hierarchical process is graphically represented as a dendrogram, or tree (see Figure 3.3).

Many criteria may be used to determine which two clusters are to be merged at each step: here, Ward's criterion was adopted, which works by minimizing the increase in the total intra-cluster variance resulting from a new merging. The initial dissimilarity matrix was built by computing the squared Euclidean distance between all pairs of activation peaks in the dataset.

[4]This can be easily inferred for SPM, as the MNI space has been adopted since the SPM96 release.

In order to get the final clustering solution, one needs to cut the dendrogram at some level and take the clusters for that level. This may be done by setting the desired number of clusters, or by establishing a threshold on some parameter of interest. Here, similarly to Jobard et al. (2003), we adopted a spatial criterion based on an estimate of the average spatial resolution of imaging experiments: accordingly, the final set of clusters was set for having an average standard deviation on the three directions not greater than 7.5 mm. Therefore, starting from the leaves of the dendrogram, our algorithm climbed up the tree and stopped just before the threshold was exceeded.

Even though hierarchical clustering is a standard and widely-used paradigm, it nonetheless suffers from the problem of non-uniqueness of the solution: it may produce different solutions depending on the order of the input data (Morgan and Ray, 1995). We therefore developed a variant of the original algorithm, whose aim was to guarantee the uniqueness of the clustering solution; this was achieved by exploring all and only the significantly different dendrograms that can be obtained starting from a given dataset (Cattinelli et al., Under Revision–b). These dendrograms are then cut at the same level (by using the above criterion based on average standard deviation), and the resulting clustering solutions were compared in order to identify the best one among them. As a quality criterion, we adopted a measure of between-cluster error sum of squares ($B - ESS$) defined as

$$B - ESS = \sum_{k=1}^{|C|} n_k \left(\mu_k - \mu_X\right)^2 \tag{3.1}$$

where $|C|$ is the number of cluster in the considered solution, n_k is the number of elements in cluster k, μ_k is the mean of cluster k, and μ_X is the mean of the entire dataset. The best clustering solution is here defined to be the one having maximum $B - ESS$ (that is to say, clusters are well-separated in the output space). The best clustering solution is then output as the final solution. This approach guarantees that the clustering result is independent from input order, up to equivalences.

Our clustering procedure identified four different clustering solutions (having $B - ESS$ values of 2.4023×10^6, 2.3977×10^6, 2.4017×10^6, and 2.3980×10^6). The final solution, corresponding to the highest $B - ESS$ value, consisted of a set of 57 clusters, having average standard deviation of 6.98, 7.34, and 7.11 mm, respectively on the x, y, and z direction.

3.2.4 Anatomical labelling

Each cluster was then automatically assigned an anatomical label based on the AAL (Automated Anatomical Labelling – Tzourio-Mazoyer et al., 2002) and Brodmann templates available under MRICro (Rorden and Brett, 2000). The mean coordinate of each cluster was mapped on these templates to extract the anatomical region to which the coordinate belongs: this was achieved in a completely automatic way through a MATLAB script that was developed to this aim. Only in those cases when the automatic polling of the templates returned a void

label[5] (i.e., when the given coordinate falls into a non-mapped brain area) the templates were manually inspected in order to find the labelled anatomical region closest to the peak in hand.

3.2.5 Anatomical consistency of clusters

Before proceeding to any further assessment of the composition of each cluster, clusters that included stereotactic peaks of both the occipito-temporal region and the cerebellum were manually split, because a cluster including both kinds of coordinates was deemed to be anatomically implausible.

3.2.6 Functional interpretation of clusters

The assessment of the functional relevance of a given cluster was performed with a preliminary qualitative assessment followed by a quantitative statistical evaluation using a binomial test.

The qualitative exploration of the data was made in order to identify any trend along two main axes: lexicality and difficulty of the materials. For example, a given cluster may have shown a clear trend for being composed of stereotactic coordinates emerging from comparisons of reading words as opposed to reading pseudowords. One such cluster would then be tested for statistical significance along the lexicality axis. On the other hand, from another cluster, a different pattern might have emerged with a substantial contribution of stereotactic coordinates derived from the comparison of pseudowords versus real words reading and irregular words versus regular words reading: such cluster would be qualitatively labelled as a brain region sensitive to the difficulty of the materials, and then subject to a binomial test along the difficulty axis in order to check for statistical significance.

Statistical testing was performed to assess the significance of the disproportion between coordinates belonging to two different, mutually exclusive categories (e.g. word-related peaks versus pseudoword-related peaks) within each cluster. If such disproportion were due to chance only, we would expect that the distribution of peaks from each category within the considered cluster would mirror the overall distribution in the dataset. Therefore, for testing the significance of a cluster along the lexicality axis, we first selected those contrasts that are word-specific (e.g. word $>$ baseline, word $>$ pseudoword), and those that are pseudoword-specific (e.g. pseudoword $>$ baseline, pseudoword $>$ word). We then computed the total number of activation peaks belonging to the two categories (words = 724 ($\approx$ 68%), pseudowords = 346 ($\approx$ 32%)). Further, if a cluster of k peaks, containing n word-related peaks was being tested for being word-specific, we applied the binomial test to assess the probability to observe at least n successes out of k trials under the null hypothesis ($\pi = 0.68$). If the cluster was instead tested for its pseudoword-specificity, the null hypothesis would have been modelled by a binomial distribution having $\pi = 0.32$. In both cases, when the P-value returned by the procedure was less or equal than 0.05, the null hypothesis was rejected.

[5]The automatic polling procedure on the AAL template returned void labels for 4 out of 57 clusters.

We proceeded similarly when testing for difficulty; in this case, contrasts were divided between those contrasting a more difficult item to an easier one, and the vice versa (for the difficulty class we had a $\pi = 0.40$; we did not test in the reversed direction).

A binomial test was used because in each comparison two classes only were considered, and because, being an exact test, the possibly scarce cardinality of a cluster would not hamper the significance of the result (as it would be the case with chi-squared test, that imposes specific requirements over the cardinality of the groups being tested). For all tests, a P-value less or equal than 0.05 was considered to be the threshold for significance.

In what follows, for illustration purposes, we also report clusters that passed only a qualitative assignment (as described earlier in this section) to one of the three classes anticipated (word-specific, pseudoword-specific, difficulty-sensitive). All other clusters were assigned to a class called *non-differentiated* (see the Supplementary Materials for further details).

3.3 Results

The procedures described left us with a final number of 64 clusters. Tables A.1 and A.2 in the Supplementary Materials describe these 64 clusters in detail.

As explained in the Materials and methods section, we first performed a qualitative analysis as in (Jobard et al., 2003) by visually inspecting the composition of each cluster, that is, the characteristics of the experiments and of the statistical comparisons that generated the activation peaks included in that cluster. This process lead us to label each cluster as (a) sensitive to item difficulty, (b) preferentially involved in word processing, or (c) preferentially involved in pseudoword processing. When none of these categories applied, the cluster was considered to be (d) non-differentiated. In what follows, we will describe these four categories of clusters; for each category, we will also indicate which clusters survived the binomial testing for significance.

A list of clusters for each category can be found in Tables 3.2, 3.3, and 3.4 (and Table A.3 in the Supplementary Materials); statistically significant clusters are marked by an asterisk. Figures 3.4, 3.5, and 3.6 (and Figure A.1) graphically show the location of the clusters within each category: a cluster is represented as an ellipsoidal blob, centred on the mean coordinate of the cluster; the standard deviation of the cluster determines the length of the semiaxes of the ellipsoid. Darker blobs indicate those clusters for which the binomial test was significant. Tables reporting information about the composition of each cluster are available upon request.

3.3.1 Difficulty-modulated network

Figure 3.4 illustrates a network modulated by task difficulty (see also Table 3.2). Within this cortical-subcortical network composed of 16 clusters, four clusters survived statistical assessment by the binomial test: the left inferior frontal gyrus (pars opercularis), the left superior parietal lobule, the right mid cingulum and the pons/cerebellum.

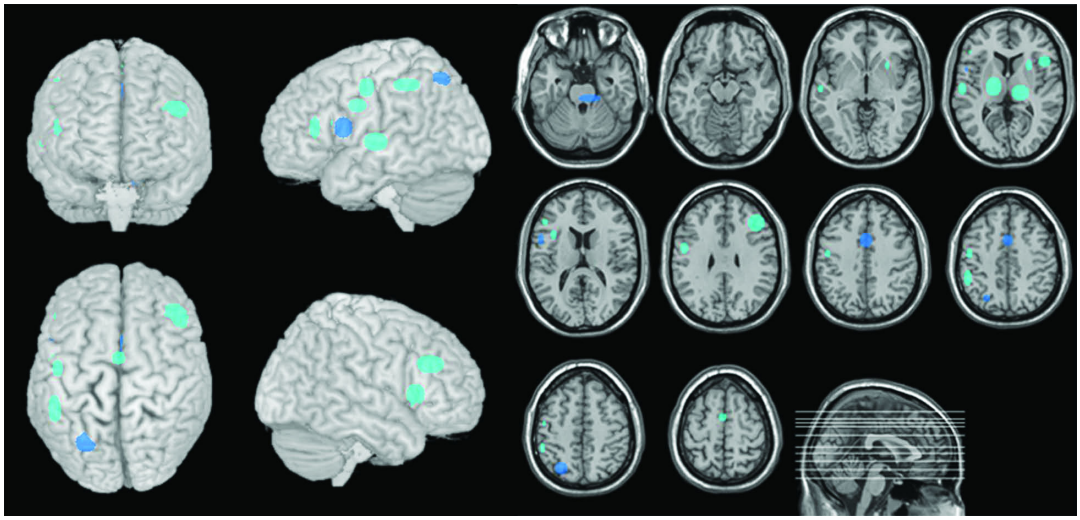


Figure 3.4: Clusters labeled as showing a difficulty effect. In darker blue, clusters surviving also statistical testing.

Anatomical Label	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*
Frontal_Inf_Tri_L_45	-46	31	14	5	17
Frontal_Inf_Oper_L	-35	15	16	6	29
Frontal_Inf_Oper_L_6*	-51	9	14	6	43
Supp_Motor_Area_L_6	0	-1	60	5	30
Precentral_L_6	-49	-2	31	6	40
Postcentral_L_6	-47	-9	45	6	29
Parietal_Sup_L_7*	-25	-66	51	7	24
Parietal_Inf_L_40	-49	-40	47	7	21
Temporal_Sup_L_22	-56	-16	3	7	29
Thalamus_L	-16	-13	6	10	31

Anatomical Label	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*
Frontal_Inf_Tri_R_45	43	30	28	10	20
Frontal_Inf_Tri_R_45	49	19	5	7	21
Cingulum_Mid_R_24*	2	8	41	8	38
Thalamus_R	18	-21	6	10	18
Putamen_R	29	14	4	7	20
Pons/Cerebellum_R*	11	-28	-25	8	8

Table 3.2: Clusters exhibiting a difficulty effect, as resulted by a qualitative assessment of their composition. Clusters for which the binomial test was significant ($p \leq 0.05$) are reported in bold and marked with a *. For each cluster, its anatomical label is reported, together with its mean coordinate, average standard deviation (averaged on the three directions), and number of included activation peaks.

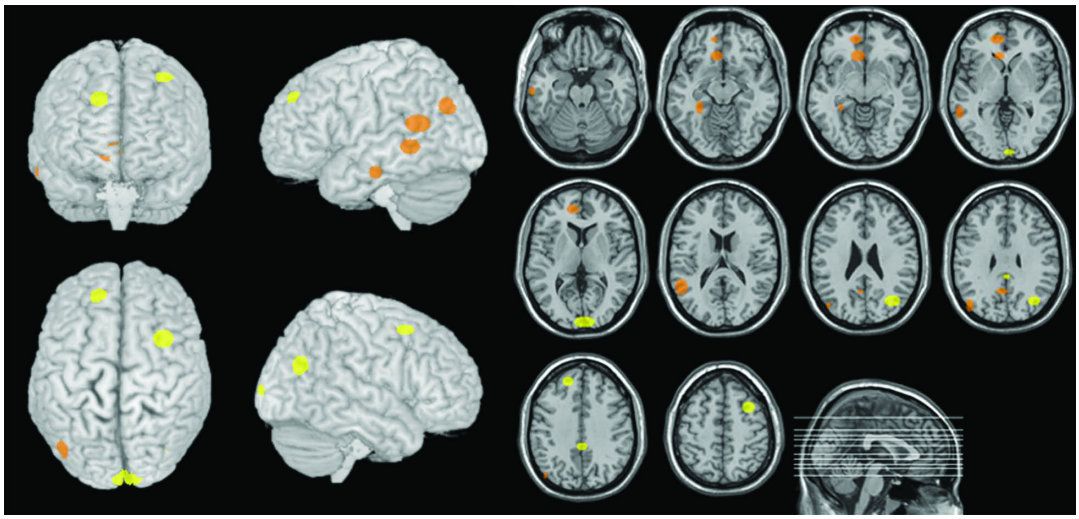


Figure 3.5: Clusters specifically involved in word reading. Clusters for which the binomial test reported a significant p-value are shown in orange.

Anatomical Label	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*	*Anatomical Label*	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*
Frontal_Sup_L_9	-15	45	36	6	10						
						Frontal_Mid_R_9	35	12	53	6	8
Cingulum_Ant_L_10*	-9	48	0	10	13						
Cingulum_Ant_L_11*	-6	27	-7	9	19						
Angular_L_39*	-44	-75	31	6	16						
						Cingulum_Mid_R_23	3	-38	38	7	14
Precuneus_L_23*	-3	-57	30	6	13						
Temporal_Mid_L_21*	-51	-51	17	8	28						
Temporal_Mid_L_21*	-57	-45	-1	6	19						
Temporal_Mid_L_20*	-61	-18	-21	5	9						
Fusiform_L_37*	-28	-39	-12	6	12						
						Occipital_Mid_R_39	38	-69	25	8	15
						Calcarine_R_17	6	-96	6	9	18

Table 3.3: Clusters showing specificity for word processing, based on our first qualitative analysis. We marked with a * those clusters for which the effect was confirmed by the result of a binomial test.

3.3.2 Word-related network

Clusters that showed a preferential involvement in word reading are listed in Table 3.3 and shown in Figure 3.5. These were mostly left-lateralized in a largely distributed network, including frontal, parietal, temporal, and occipital regions.

Over a total of 13 clusters, 8 survived statistical testing; all of them are located in the left hemisphere. Two clusters fall in the anterior cingulum; two clusters are located in the parietal cortex (respectively, in the posterior part of the angular gyrus and in the precuneus); three clusters can be found in the middle temporal gyrus; and one final cluster is in the anterior part of the fusiform gyrus.

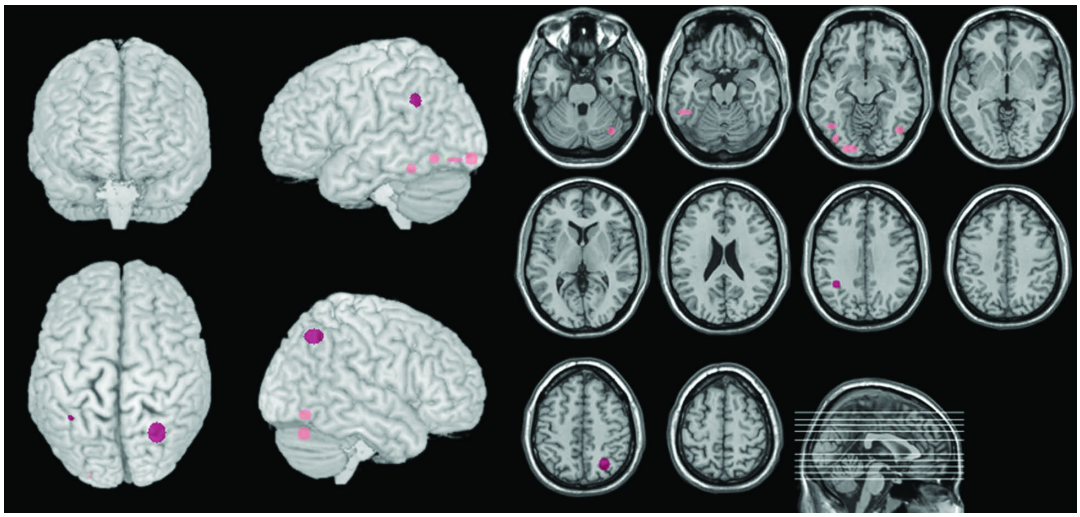


Figure 3.6: Clusters showing a specificity for pseudoword processing. Among these clusters, those having a statistically significant effect are depicted in purple.

Anatomical Label	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*
Angular_L_40*	-35	-49	34	6	21
Temporal_Inf_L_20	-48	-46	-19	6	17
Fusiform_L_19	-36	-80	-12	4	5
Occipital_Inf_L_18	-18	-93	-11	7	19
Occipital_Inf_L_37	-41	-64	-11	4	26

Anatomical Label	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*
Angular_R_7*	30	-62	49	7	28
Occipital_Inf_R_19	46	-69	-12	5	10
Cerebelum_Crus1_R	39	-70	-27	5	7

Table 3.4: Clusters specifically involved in pseudoword reading, according to our preliminary qualitative inspection. Clusters that survived statistical testing are shown in bold and marked with a *.

3.3.3 Pseudoword-related network

Eight clusters were classified as being preferentially involved in pseudoword reading (see Table 3.4 and Figure 3.6). The pseudoword network encompasses bilateral parietal areas, mostly left-lateralized temporo-occipital regions, and the right cerebellum.

Two individual clusters survived a direct statistical assessment. These are located in the right angular/superior parietal gyrus and in the left inferior parietal lobule, in the supramarginal gyrus. Because of its well-known involvement in orthographic processing, the set of the four left ventral occipito-temporal clusters was also analyzed collectively, yielding a significant pseudoword effect; this joint analysis was equivalent to an ad-hoc change in spatial resolution.

3.3.4 Non-differentiated clusters

The remaining 27 clusters, which could not be assigned to any of the previous categories, were considered to be non-differentiated clusters. This generic network includes bilateral frontal regions, right parieto-temporal areas, bilateral fusiform gyri, the right hippocampus, bilateral occipital cortex and cerebellum. For further description of these particular results, the interested reader can refer to the Supplementary Materials.

3.4 Discussion

Our meta-analysis identified three reading systems: a network sensitive to the difficulty of the materials, a network specifically involved in word processing and a network preferentially involved in pseudoword reading. In addition, we also found a set of clusters that somehow participate to reading processes, but to which it was impossible to attach a specific functional commitment, even in a merely qualitative way.

In what follows we will discuss these results in the light of previous review articles or in the light of data interpretations offered in notable single studies on reading; we will propose a comparison of our study to a similar meta-analysis on reading, the one published by Jobard and colleagues (2003). Additionally, implications for existent cognitive models of reading will be discussed. Finally, we will consider recent literature that has been published after our analysis was completed, in order to assess whether new results are consistent with what emerged from our study. Citations of neuropsychological data complement our discussion.

3.4.1 Difficulty-modulated network

The clusters assigned to this network are characterized by a predominant number of activation peaks coming from statistical comparisons where a more difficult item type is contrasted to an easier one; for instance, *pseudoword* $>$ *word* (difficulty in a lexical sense), *exception word* $>$ *regular word* (difficulty in a spelling-to-sound-consistency sense), *low-frequency word* $>$ *high-frequency word* (difficulty in a frequency sense). We propose that these brain regions are sensitive to the computational load required by the reading task in hand, rather than to any of the considered psycholinguistic variables per se (lexicality, or consistency, or frequency); a plausible interpretation for this effect is, in fact, that in these regions difficult tasks induce longer processing times that, as a consequence of time integration, result in stronger BOLD signal.

There were four such clusters: one in the left inferior frontal gyrus (pars opercularis), one in the (right) mid cingulum, one in the left superior parietal gyrus, and one in the pons.

In the reading literature, an important role is attributed to the left inferior frontal gyrus (pars opercularis), long considered to be part of the output system since Broca's study (1861). In their review, Fiez and Petersen (1998) discuss the role of the left frontal operculum in transforming orthographic representations into phonological ones, and point out its sensitivity to consistency and frequency of written stimuli: this area appears to be more activated by low-

frequency exception words and by pseudowords, rather than by regular words[6]. The role of the left frontal operculum in phonological processing, and its relative larger activation for pseudowords rather than for words is recognized also in (Price et al., 2003), with complementary evidence coming also from patients' data. They described a case of semantic dementia associated with surface dyslexia, investigated through an fMRI experiment: in this patient, reading elicited a greater activation of the left opercular frontal region, as compared with normal subjects; this may suggest that, being the semantically-mediated route disrupted by neural atrophy, a surviving phonological route was stressed in an attempt of compensation.

An association between a lesion of the left frontal operculum and an effect that can be interpreted along a difficulty axis was also found in a multiple subjects lesion study by Fiez et al. (2006). The group of 11 patients with a left frontal operculum lesion showed a significantly greater impairment in reading pseudowords than regularly spelled words; however, the same patients had troubles in reading low-frequency irregular words. This pattern may be unusual for phonological dyslexia as a syndrome, for which only a pseudoword deficit is usually described (Rapcsak et al., 2009): for this case it may be useful to escape from a syndromic definition of the dyslexia under investigation and to rather concentrate on the anatomical underpinnings and their potential meaning. In this perspective, the study by Fiez et al. (2006) suggests that alexia due to a lesion centred on the left inferior frontal gyrus brings about a syndrome due to a general deficit in phonological output processes, and not only in the transformation from sub-lexical orthographic to phonological representations; further support for this proposal is provided by Bird et al. (2003) with respect to paste-tense morphology. The effects of such a deficit would be more evident as task difficulty increases, as predicted by the analysis of the cognitive components involved in reading according to a dual-route perspective (see Figures 1 and 2), but also in a connectionist "triangle" model perspective.

Our interpretation about the left frontal operculum contribution in reading is also shared by others (Fiez et al., 1999; Mechelli et al., 2005). Mechelli et al. (2005) suggest that *"this region may contribute to the effortful selection, retrieval, and/or manipulation of phonological representations critical for reading and other phonological tasks"*.

On the other hand, Binder et al. (2005) offer a collective - i.e. not limited to Broca's area - interpretation of a putative "difficulty network", as the activity within this network correlated with the reaction times for their reading task. In their words *"... in both cases"* - i.e. reading pseudowords and low-frequency irregular words - *"the relative unfamiliarity of the particular correspondence makes the mapping process less efficient, resulting in an increased load on attentional (FEF, IPS, anterior cingulate), working memory (IFG, precentral gyrus, IPS), decision (IFG, anterior cingulate gyrus, anterior insula), and response monitoring (anterior cingulate gyrus) mechanisms. Activation of these regions is therefore consistent with the expected and observed differences in task difficulty between conditions, and not indicative of a specialized route for rule-based phonological assembly"*.

[6]Based on this observation, (Price et al., 2003) also suggest that there might be a link between left frontal damage and phonological dyslexia (a syndrome characterized by a specific impairment in pseudoword reading), although only a half of the cases they reviewed were actually associated with frontal lesions.

The desire to provide a comprehensive interpretation of the difficulty network is a seducing one but it should not hamper the possibility of making a finer grained analysis of the individual regions involved in the network. This is the position we favour.

As illustrated in Figures 3.1 and 3.2, for example, both irregular words and pseudowords should stress to a greater extent phonological output processes when compared with regular words: the left inferior frontal region is an excellent candidate for such processes. Much less is the case for the left superior parietal lobule or for the cingulate region. These may participate to the difficulty network for other reasons than because involved in phonological output processes.

The parietal lobe and the anterior cingulate cortex have been implied in attentional processes with and without explicit eye-movements (reviews in Bottini and Paulesu, 2003; Corbetta, 1998; Kastner and Ungerleider, 2000). Neuroimaging studies consistently found attention-related activation in the superior parietal lobule, frontal eye field, and supplemental eye field (Corbetta, 1998; Kastner and Ungerleider, 2000). These findings therefore provide support to the interpretation of the superior parietal cluster and mid-cingulum cluster as involved in attentional processes, whose contribution is presumably modulated by task difficulty.

To summarize the results on the difficulty network, previous evidence, and our own, both support the hypothesis that the left inferior frontal gyrus, pars opercularis, participates in translating orthographic representations into phonological ones, a process that is sensitive to difficulty. The mid-cingulum cluster and the left superior parietal cluster might reflect attentional processes that are more vastly recruited for more effortful stimuli, such as exception words and pseudowords, that require more cognitive resources. It may be argued, especially for the superior parietal region, that the difficulty of the stimulus triggers a more intense eye movement activity for extensive exploration of the orthographic string. Interestingly, a similarly placed cluster in the right hemisphere was also present for the pseudoword-sensitive network.

Lastly, the cluster whose centre is located in the dorsal part of the pons could tentatively be related to the ascending modulatory pathways of the reticular formation involved in attention. Yet, the small number of activation peaks (eight) grouped in this cluster invite to treat this finding with caution.

3.4.2 Word-related network

Our meta-analysis also identified a set of left brain regions showing a significant preference for real words as opposed to pseudowords. These areas form a temporoparietal network including the posterior part of the angular gyrus, the precuneus, the middle temporal gyrus, and the anterior part of the fusiform gyrus; in addition, two clusters in the left anterior cingulum were found.

Words differ from pseudowords both because they are familiar items, and because they possess semantic value. If both whole-word orthographic forms and semantic forms are sep-

arately stored somehow in the cortex (as implied by dual-route accounts of reading based on patients observations), then it should be possible to find, among these word-specific areas, both semantic and purely lexical regions. This is what is implied by the subtractive logic of cognitive neuropsychological studies in which the co-occurrence of semantic deficits and surface dyslexia is denied for cases with a specific surface dyslexia (Coltheart et al., 1983) (see however note 3).

However, the evidence from previous imaging literature collectively suggests that the word-sensitive areas that we identified in this work are in fact involved in semantic processing as well. Indeed, Price and Mechelli (2005) report that the left angular gyrus, middle temporal cortex, anterior fusiform gyrus, as well as the left inferior frontal gyrus in the pars orbitalis and triangularis are consistently activated in semantic tasks as opposed to phonological tasks.

The panoramic description of the semantic network given by the meta-analysis of Binder and colleagues (2009) is consistent with our findings. All the brain regions that we associate with a word-specific reading network belong to a semantic system. Thus, there is evidence that our word-sensitive areas most likely reflect semantic processing, whereas it is not possible, on the basis of the available data, to identify any purely lexical region.

Additional insights may be drawn from literature on reading disorders. In (Brambati et al., 2009) a VBM (voxel-based morphometry) analysis was performed on a group of 54 subjects, including both normal subjects and patients suffering from different forms of neurodegenerative disease. Correlations between gray matter volume and reading accuracy, with respect to exception words and pseudowords respectively, were computed. This analysis revealed that a positive correlation exists between accuracy for exception word and gray matter volume in the left anterior middle and superior temporal gyri, in the left temporal pole, and in the anterior part of the fusiform gyrus.

It should also be noticed that the foci of lesion observed in most of the surface dyslexic patients described by Vanier and Caplan (1985) is largely overlapping with our word-related network. In light of what we have discussed above, and of the frequent co-occurrence of semantic deficits in cases of surface dyslexia (Patterson and Lambon Ralph, 1999), this observation strengthens the idea that our word-related clusters are indeed part of the semantic network.

Our findings on the word-related network present a substantial difference as far as the frontal regions, particularly for the anterior and orbital parts of the left inferior frontal gyrus. Perhaps this is not that surprising if one considers the rather long list of effects/functions that have been associated with the inferior frontal cortex (for examples on this point, see Cortex, vol.42 (4), 2006, pp. 461-658).

Indeed, while there is a broad consensus that the left inferior frontal gyrus should show some degrees of specialization, the details are still a matter of debate. According to some, the more anterior and ventral aspect of the inferior frontal gyrus (BA 47/45) should be involved in semantic processing, the more dorsal and posterior part (BA 45/44) being involved

in phonological processing (Poldrack et al., 1999). This proposal is similar to the earlier one of Paulesu et al. (1997).

However, none of the above mentioned fractionations of the left inferior frontal gyrus can be easily reconciled with our meta-analysis data on reading, with the notable exception of the opercular inferior frontal gyrus if this is to be associated with phonological output processes. Indeed, our meta-analysis associates BA 44 and 45 (45 in a qualitative manner), to the difficulty network, while the orbital part of the inferior frontal gyrus (BA 47) was definitely non-differentiated as far as lexicality, regularity or frequency of the items in hand.

Idiosyncratic factors due to the nature of reading may contribute to make a comparison of the reading data with the non-reading language literature difficult at times, at least as far as the frontal cortex is concerned. The superimposition of task demands over those due to the structure of the stimuli may contribute to this riddle.

Other possible factors may be due to a faulty conceptualization of which neural/cognitive components should operate when dealing with certain kinds of stimuli during reading: for example, the concept that pseudoword reading should involve only sub-lexical non-semantic rule-based reading procedures[7].

3.4.3 Pseudoword-related network

Finally, there were clusters having a reliable pseudoword effect: one located in the left inferior parietal gyrus, and one in the right parietal cortex, between the angular and the superior parietal gyrus.

The left inferior parietal cluster associated with pseudowords falls in a region frequently associated with sub-lexical phonological processes for tasks not necessarily involving reading (Démonet et al., 1994; Paulesu et al., 1993, 1996), at the border between the supramarginal and the angular gyrus. This region falls anteriorly to a word-related angular gyrus cluster. Identification of two functionally different and separate clusters within the same larger inferior parietal regions helps to clarify a riddle about the role of inferior parietal cortex, particularly with reference to the angular gyrus. The association of the angular gyrus to reading is as old as neuropsychology, starting from the observations of Dejerine on alexia with agraphia (Dejerine, 1892). In this sense, the fact that our meta-analysis found two clusters in this region, with a different functional attribution (semantic processing vs. pseudoword reading) might suggest that the historical controversy about the role of the left angular gyrus in reading is, at least to some extent, due to a specialization of this brain area into separate neural subsystems.

On the other hand, as already mentioned, the position of the right parietal cluster mirrors another left parietal cluster associated with the difficulty effect nearby the parietal regions involved in eye movement control. A similar functional interpretation can be given for this right sided region: the complexity of the input stimulus may trigger more attentional resources, with particular reference to the exploratory eye movements necessary to analyze the structure of the stimulus.

[7] For more comments on this point, see below in the discussion about neurocognitive models of reading.

While the two parietal clusters were the only ones to survive statistical testing individually, four left ventral occipito-temporal clusters, once analyzed collectively, showed a significant pseudoword preference. This suggests the existence of a more distributed network, subsuming all these clusters, to which the pseudoword effect can be attributed.

A pseudoword network partially similar to the one found here has been proposed in (Brambati et al., 2009): their correlational analysis between pseudoword reading accuracy and gray matter volume was significant for the left inferior parietal lobule (angular gyrus), the posterior aspect of the middle and superior temporal gyri, and the posterior fusiform gyrus.

The ventral occipito-temporal network encompassed by the four clusters that were analyzed collectively includes the midfusiform region known as Visual Word Form Area (VWFA; Cohen et al., 2000, 2002). This region, centred on Talairach coordinates x = -43, y = -54, z = -12, has been interpreted as being dedicated to the analysis and recognition of orthographic forms for visually presented letter strings. Currently, there is some consensus on the view that this region shows a functional specialization, although not exclusive, to the orthographic analysis of units of written material (from single letters up to, possibly, entire words). However, a controversy exists in the literature about the putative role of the VWFA. Price and Devlin (2003) review neuropsychological and neuroimaging evidence against this interpretation. From a neuropsychological perspective, they report that no one-to-one correspondence has been found in literature between pure alexia (specific reading deficit, not accompanied by agraphia or by oral comprehension disruption) and focal lesions to the WVFA. As for the neuroimaging evidence, Price and Devlin (2003) show that this midfusiform region is actually activated also in tasks that do not involve the presentation of orthographic stimuli, such as picture naming (and to a greater extent than word reading itself), and tasks using auditory material. It has been proposed (Price and Mechelli, 2005) that a more plausible interpretation of the role of the VWFA is that of an interface for the retrieval of phonology from a visual input.

Both positions (in favour or against the existence of a visual word form area) agree in that the putative computation taking place in the VWFA is not exclusively committed to either words or pseudowords, but is an early, shared step in reading associated with orthographic input processing.

However, a pseudoword effect here remains plausible if one considers that pseudowords may be more resource demanding if representations are held here of orthographic strings of a grain size up to that of a whole word. Incomplete matching of a pseudoword with pre-existing templates may lead to extensive search through the neuronal representation, hence, larger activations in the comparison of pseudowords versus words (cf. Paulesu et al., 2000).

It should also be noted that the different subdivisions of the left fusiform gyrus have shown some degree of specificity to either word or pseudoword processing. In particular the most posterior part of the fusiform gyrus has already been associated with pseudoword processing in a number of studies (see, for instance, Mechelli et al., 2005), as opposed to the

more anterior aspect that is reported to be more consistently activated by semantic tasks. The VWFA lies in between these subdivisions. Due to the necessarily limited resolution of our meta-analysis, it cannot be guaranteed that such a fine distinction about fusiform subdivisions is preserved, and therefore pseudoword-related activations that might be ascribed to the more posterior part, might confound the picture regarding the VWFA. However, our data did capture the anterior-posterior functional distinction for the left fusiform gyrus, with the posterior fusiform cluster being classified among pseudoword-related areas (once the significance was assessed collectively across four clusters), and the anterior fusiform cluster entering the word-related (semantic) network.

In addition, it is worth noting that among the ventral inferior temporal cortices associated with pseudoword reading lies the so-called lateral-inferotemporal multimodal area (LIMA), an area of overlay between activations for reading or for phonological processing (Cohen et al., 2004), and therefore a region where an initial integration between orthography and phonology may take place.

3.4.4 Results from more recent studies and other meta-analyses

The process of performing a meta-analysis in a lively area of research such as the one of reading has the inevitable drawback that by the time that the analysis is completed and the ensuing paper is written, new data have appeared in the literature. We turn this potential issue[8] into an opportunity by reasoning that a valid[9] meta-analysis should hold predictive power when faced with subsequent studies. Indeed, a comparison with newer findings should offer a good testbed for the meta-analysis itself. Here, we will discuss some studies that have appeared in the literature between second half of 2008 and the first trimester of 2010, selected on the basis of the same inclusion principles that guided our meta-analysis and with a preference for those studies investigating specific sub-processes of reading. The examination of this literature started in March 2010 while all analyses for this paper were completed by December 2009.

As our meta-analysis is based on papers that adopted univariate analyses of the data, we do not consider here articles based on analyses such as effective connectivity or multivariate approaches. For the effective connectivity papers (e.g. Levy et al., 2009) we remark that these assessed only a limited number of cortical nodes making it impossible to fully test competing cognitive models of reading.

Following these criteria, we isolated the following additional papers: Graves et al. (2010); Hauk et al. (2008); Levy et al. (2008); Seghier et al. (2008); Kronbichler et al. (2009); Carreiras et al. (2009); Peeva et al. (2010); Nosarti et al. (2010).

For the sake of brevity, we will discuss in detail only one of these studies (Graves et al.,

[8]We hope that the reader will appreciate that it is practically inevitable that a meta-analysis is submitted without incorporating all published papers on a given topic, particularly those published just before a submission. If meta-analyses were meant to capture a general trend of a biological principle, the eventual new manifestation of that principle should not change the overall understanding about it.

[9]The validity of a meta-analysis, of course, much depends on the richness of the data under investigation.

2010); for the remaining ones we will make a summary comment. We chose to focus on (Graves et al., 2010) because of its psycholinguistically detailed experimental design. This study, as many before, used fMRI to identify systems involved in orthographic, phonological, and semantic processing in reading aloud; however, the stimuli were selected in such way that the six factors of interest (frequency, consistency, imageability, bigram frequency, biphone frequency, and length in letters) were kept uncorrelated. Linear regressions were calculated between the BOLD signal and these variables or with the reaction times. Interestingly, reaction times were positively correlated with activity in the left inferior frontal gyrus and anterior insula, and the same areas also showed a negative correlation with word frequency and consistency. These areas were interpreted to be involved in generic (i.e. not specific for reading) executive processes, sensitive to increases in task load. This is compatible with our proposal that the left frontal operculum contributes to reading in a difficulty-modulated manner, although we also propose that its involvement might be specific to phonological output processes, rather than just due to more general attentive, working memory, or executive processes. A positive correlation with reaction times paired with negative correlation with frequency and consistency was observed in the left midfusiform gyrus (VWFA), for which a possible role in the mapping from orthography to phonology was suggested.

Positive correlations with word frequency and imageability were found bilaterally in the angular gyrus and precuneus/posterior cingulated cortex, suggesting a semantic role for these areas. Semantic involvement is also proposed for the left middle temporal gyrus and inferior temporal sulcus, on the basis of their negative correlation with word consistency. In this case, the engagement of the semantic system would be specifically aimed at helping the pronunciation of inconsistent words. A similar pattern of activation was observed in the left inferior frontal gyrus, pars orbitalis and triangularis, for which a role in modulating attention in the semantic system (MTG/ITS) is proposed. The involvement in semantics for the left angular gyrus, precuneus, and middle temporal gyrus is supported also by our meta-analysis.

Finally, negative correlations with bigram frequency were found in bilateral posterior middle temporal gyrus, superior temporal sulcus, and left supramarginal gyrus; this was interpreted as an effect of increasing difficulty in the direct mapping (i.e., not semantically mediated) from orthography to phonology. The left supramarginal gyrus location described by Graves et al. (2010) is sufficiently close to our left inferior parietal cluster to consider this as a possible replication.

As for the results reported by the other recent studies listed above, there is further evidence for a difficulty effect, reliably localized in the left inferior frontal gyrus. Hauk et al. (2008) found a negative correlation with frequency for silent word reading in the left inferior frontal gyrus. This finding was confirmed in (Carreiras et al., 2009), where activation in frontal regions (specifically, left pre/SMA and inferior frontal gyrus/insula bilaterally) was negatively correlated with word frequency for a lexical decision task. In (Levy et al., 2008) a precentral gyrus focus close to our left frontal operculum cluster was activated for the comparison pseudowords $>$ words, which contributes to a difficulty effect (as acknowledged in

this particular study, too, as pseudowords are the type of stimulus involving the greatest processing load). Also Kronbichler et al. (2009) reported foci in the inferior frontal gyri and left precentral gyrus for pseudoword (letter-deviant versions of real words) reading as opposed to word reading. Peeva and collaborators (2010) explicitly interpreted their left frontal operculum focus of activation as a difficulty effect. Lastly, Nosarti et al. (2010) again identified the left frontal operculum as part of a network involved in processing load (*"when nonlexical and lexical/semantic processing are inconsistent"*).

We have also found replications of parts of what we identified as a semantic network. In (Seghier et al., 2008) activation of an anterior occipito-temporal region correlated with off-line performance in irregularly spelled word reading; the correlation had a negative sign in that the more the disadvantage for irregular word reading, the more the activation in the anterior occipito-temporal region. The interpretation of this finding offered by the authors is that more neural labour is required in this region by those who have more limited skills for irregular word reading (the same interpretation for the pseudoword reading effect found in our meta-analysis and, previously, for cross-cultural differences in pseudoword reading (Paulesu et al., 2000)). Seghier et al. (2008) view is compatible with the (lexico-)semantic role for the anterior occipito-temporal region that we hypothesize in our meta-analysis. The *words* $>$ *pseudowords* comparison reported in (Levy et al., 2008) identified a focus of activation in the bilateral inferior parietal lobule; the location of this left parietal focus is close to our word-related left angular cluster. The lexical decision task used in (Carreiras et al., 2009) revealed greater activation for high-frequency words versus low frequency words in the precuneus; this result is consistent with our classification of this area in the word-related (semantic) network.

Finally, additional data have been found that are concordant with our pseudoword-specific network. In (Levy et al., 2008), pseudoword reading yielded greater activation in a left parietal region close to our pseudoword-specific cluster in the same region, and in the left occipito-temporal area, in agreement with our findings; for these regions a role in orthographic processing is proposed. Seghier et al. (2008) reported a negative correlation between pseudoword reading performance and activation in a network involving the left posterior occipito-temporal region and bilateral intraparietal cortex, suggesting that an ineffective pseudoword reading strategy would increase the processing load on such areas. These areas are compatible with our pseudoword-related network. Activation in the left occipito-temporal region for pseudoword reading was found also in (Kronbichler et al., 2009) and (Nosarti et al., 2010).

What remains missing is an explicit demonstration of a dedicated set of regions for the so-called orthographic input lexicon.

To complement our review of the literature, a brief discussion of other meta-analyses on the neuroimaging literature of reading is in order: the papers by Turkeltaub (2002), Jobard et al. (2003), Tan et al. (2005), and Bolger et al. (2005). These are discussed more extensively in the Supplementary Materials section. We group them here according to the technique adopted.

Using the Activation Likelihood Estimate approach - ALE - or variants of it[10], Turkeltaub (2002), Tan et al. (2005), and Bolger et al. (2005) assessed the existence of areas of consistency in the activation patterns across studies, or commonalities and discrepancies among different orthographic systems (western alphabetic versus Chinese or Japanese). In brief, these studies cannot be compared with our own as they lack an explicit assessment of the functional role of the clusters of activations. In addition we concentrated on alphabetic orthographies only.

The meta-analysis by Jobard et al. (2003), while more similar to our own as they used a hierarchical clustering technique, is different in several respects: we adopted stricter inclusion criteria for building our dataset; we developed an improved clustering method that controls for multiple solutions, and we used statistical testing for assessing the functional role of clusters, rather than providing a qualitative evaluation[11]. Besides, while Jobard and collaborators (2003) explicitly classified statistical comparisons of the imaging data according to a dual-route model of reading, we performed a more theory-independent classification. For this reason, comparing our conclusions with results in (Jobard et al., 2003) may not be possible or informative: yet we believe that our effort represents a step forward for the above mentioned reasons. In addition, we note that while Jobard et al. (2003), on the basis of a qualitative assessment of the data, claimed to have evidence in support of a dual-route model for reading, our claim is substantially different in that we propose that a connectionist single-mechanism model can account for the data while a dual-route account would require a number of modifications, some of which appear anatomically or functionally implausible when challenged with patients' data (see the final section of the Discussion). Furthermore, in (Jobard et al., 2003) the left frontal operculum is classified as part of the GPC route, while in our analyses this appears to be a brain region sensitive to the difficulty of phonological retrieval/processing.

3.4.5 Towards a functional model of reading

In this final section we sketch a functional model of reading that takes into account both the results of our meta-analysis and the discussion entertained in the previous pages, crucially including also data from patients. The model is illustrated in Figure 3.7.

We propose that the ventral occipital and occipito-temporal regions (pink blobs in Figure 3.7) may be a first neuronal station where visual information is routed to after early visual processing. Here, orthographic processing takes place, and possibly there is a first stage of orthographic to phonological integration, particularly in the more anterior and lateral part of this set of areas, in a region roughly corresponding to the area called LIMA (Cohen et al., 2004). Further evidence of some degree of overlap between a phonological task and a reading task has been recently found in this same region (Danelli and Paulesu, personal communication).

[10] Bolger et al. (2005) used the AGES (aggregated Gaussian-estimated sources) approach.

[11] In (Jobard et al., 2003) some functional assignations to one of the two routes for reading were made even when the number of activation peaks from a class of statistical comparisons (e.g. "indirect route" peaks) were in an equal (or very close to equal) number as those from the other route.

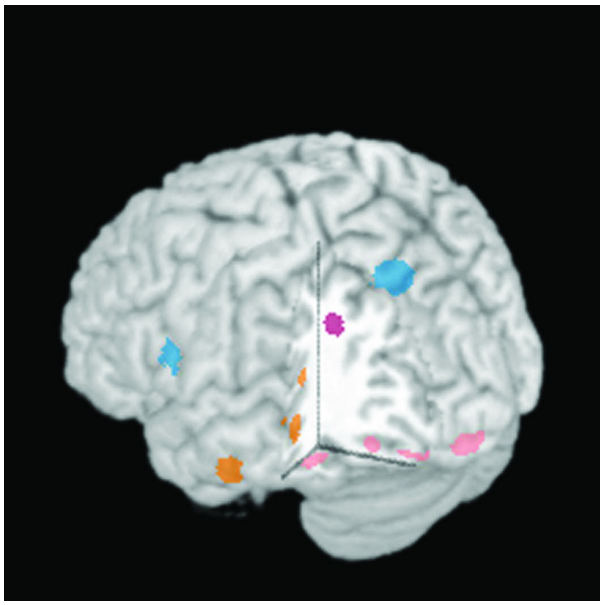


Figure 3.7: A hypothetical model of the reading network based on the results of this meta-analysis. No explicit connections are represented as these are not evaluated in this study. In pink, orthographic input nodes collectively more active for pseudoword reading, but also involved in word reading; in orange, word-specific nodes also involved in semantic processes; in purple, a region associated with pseudoword reading; in blue, areas sensitive to items difficulty, with the left inferior opercular region associated with phonological output processes. For a full description, see the final section of the Discussion.

The same set of clusters also contains the so-called VWFA, whose role is more controversial: according to Cohen and colleagues, the region is specific for orthographic representations; according to others (e.g. Graves et al., 2010; Hillis et al., 2005; Price and Devlin, 2003) it might have a an input-output integrative function, while not being committed to orthographic stimuli only. Even though the ventral occipito-temporal regions tend to respond more strongly to pseudowords than to words, this stage of orthographic processing appears to be common to words and pseudowords.

Interactions of these occipito-temporal regions with the word-semantic network (orange blobs in Figure 3.7) may contribute to complete the extraction of word phonology for real words, particularly for words with irregular spelling. The location of the word-semantic network identified here is consistent with the dominant pattern of brain damage seen in patients with surface dyslexia (Vanier and Caplan, 1985).

On the other hand, during pseudoword reading the left supramarginal gyrus (purple blob in Figure 3.7) may contribute either to further integration between phonological knowledge and orthographic representations, or to phonological implementation in speech output.

Semantically-mediated and more phonologically-mediated mappings are then routed to the frontal lobe (blue blob in Figure 3.7) for further processing: the final pronunciation may be decided here, by means of integrative and competitive mechanism.

Some predictions can be made on the kind of neuropsychological deficits that would be

observed when damaging any of the components of the proposed model. In turn, we also discuss which of these predictions are fulfilled by available data on patients.

A left occipito-temporal damage would impair basic orthographic processing, thereby resulting in letter-by-letter reading, a suggestion consistent with classical anatomical findings in alexia without agraphia (see review in Cappa and Vignolo, 1999) and recent subdural electrical inhibition data (Mani et al., 2008). This observation supports the idea that both real words and pseudowords are necessarily processed here in a normal fluent reader.

A left inferior parietal lesion, or a damage causing a disconnection between the ventral occipito-temporal cortex and this site, would hamper aspects of the grapheme-to-phoneme mappings or phonological implementation; this would be especially detrimental to pseudoword reading, since these cannot rely on semantic mediation to recover phonological codes. Such a lesion may therefore produce a pattern of phonological dyslexia, not deprived, though, from other phonological symptoms. However, lesions of the left supramarginal gyrus are not associated with a pure phonological dyslexia; rather, patients with such lesion typically show either conduction aphasia (Damasio and Damasio, 1980) or phonological short-term memory deficits (Shallice and Vallar, 1990; Vallar et al., 1997). The former syndrome is associated with errors in pseudoword reading (Bisiacchi et al., 1989); however, errors with pseudoword processing are not limited to reading tasks. To make matters more complicated, there are patients with a lesion involving the left supramarginal gyrus and phonological short-term memory deficits who could read pseudowords (T. Shallice, personal communication about patient JB; G. Vallar personal communication about patient PV).

On the other hand, if the semantic network is damaged, then surface dyslexia would be expected: the semantic contribution to reading would be compromised, and this could hamper especially the reading of irregular words, that rely more than regular words on semantic mediation. Surface dyslexics patients, who, according to a dual-route perspective, should suffer from a deficit of the orthographic lexicon, typically have lesions of the middle temporal gyrus/superior temporal sulcus or of the angular gyrus (Vanier and Caplan, 1985). This is not where we could possibly locate orthographic representations deprived of semantic content (the VWFA and neighbouring regions) even according to a modified dual-route model described above.

Finally, a lesion in the left frontal operculum would result in reading deficit especially for the most difficult stimuli, such as pseudowords and irregular words (Fiez et al., 2006), rather than in a reading deficit sensitive only to the lexical status of the stimuli.

To what extent is our model consistent with classical cognitive models of reading like the dual-route theory (Coltheart et al., 1993, 2001) or the connectionist single-mechanism accounts (e.g. Plaut et al., 1996), to cite the most popular ones?

Let us first consider the dual-route theory (Fig. 3.1; Coltheart et al., 1993, 2001): this postulates the existence of a non-semantic lexical representation. As discussed in the Introduction, using functional imaging, such area could be isolated paradoxically, if a modified account like the one illustrated in Fig. 3.2 is assumed, by comparisons of pseudoword reading versus

word reading; however, the candidate brain region for such a level of representation is in the ventro-occipital network, well outside the main lesion pattern of surface dyslexics (Vanier and Caplan, 1985); in addition, the region responds more powerfully for pseudowords, a finding that requires a reconsideration of the role of large-sized orthographic representations in pseudoword reading, bringing perhaps new blood to the old-fashioned reading by analogy model (Kay and Marcel, 1981).

Another fact that militates against a dual-route model for reading is that a dedicated grapheme-to-phoneme conversion system could not be identified either: some of the regions (e.g. the left supramarginal gyrus) can be associated with sub-lexical phonology also for auditory language processing; some are most likely associated with attention or eye-movement control that we assume to be more stressed in pseudoword reading.

A connectionist single-mechanism model, like the triangle model (Plaut et al., 1996) – (see also Seidenberg and McClelland, 1989), fares perhaps better: this model has the natural appeal of making fewer assumptions in terms of modularity of the various components involved in an acquired skill such as that of reading; the model also does not assume an explicit and unique dedication of the components to the process of reading. Because of the fewer assumptions, the model is not hampered by the observation of concurrent deficits in the domains of visual perception, semantics or phonology when explaining syndromes such as letter-by-letter reading, surface or phonological dyslexia (see, for example, Patterson and Lambon Ralph, 1999). More specifically, the main tenet of the PMSP model is that distributed orthographic, phonological, and semantic codes interact to produce skilled reading of all kinds of written stimuli. Accordingly, there is no need for two separate and computationally different routes to process exception words and pseudowords. However, even in this model two pathways can be discerned: one that directly maps orthographic codes into phonological ones, and one that is also mediated by semantic influences. Reading (and learning to read) occurs, according to the division of labor hypothesis (Plaut et al., 1996; Harm and Seidenberg, 2004), by cooperation of the phonological and the semantic pathways: since the semantic pathway intervenes to support naming for words only, the phonological pathway learns to count on such contribution and experiences less pressure to perfectly master all words. As a result, the contribution of the semantic pathway tends to get greater for "hard" words (irregular and/or low frequency items), whereas the purely phonological pathway gets to be more finely tuned to pseudoword reading (which, on the other hand, cannot take advantage of any semantic information).

All these features are supported by the data. No explicit demonstration exists of a region that has representations of the size of a non-semantic orthographic lexicon; the brain damage best associated with surface dyslexia is in regions involved in semantics; sub-lexical phonological areas are not merely involved in reading.

Although in principle acceptance of some of the above evidence would correspond to acceptance of a series of null hypotheses, we cannot but acknowledge that a connectionist model has more evidence than an explicit dual-route model for reading from our meta-analysis.

3.5 Conclusion

In this chapter a meta-analysis on neuroimaging studies investigating reading processes has been presented. A word-specific, pseudoword-specific, and difficulty-sensitive set of brain regions were identified. Evidence from the literature supports the interpretation that the word-related network may coincide with the neural system involved in semantic processing, and the pseudoword-related one may be involved in the direct mapping of orthographic representations into phonological ones. The difficulty effect, especially as far as the left frontal operculum is concerned, may be interpreted as the result of competition among discrepant, or unclear, phonological responses that arise when the mapping from orthography to phonology is a difficult one (either for reasons of consistency, frequency, or lexicality). The results discussed here provide some support for a connectionist distributed model for reading.

On the other hand, the attempt to fit the imaging data with a dual-route account of reading clashed against a number of facts, the primary ones being the inconsistency with patients' data.

Part II
Exploring cognitive modelling

Chapter 4

The quest for understanding the human mind, part 2: the computational modelling way

"My mind seems to have become a kind of machine for grinding general laws out of large collections of facts."
— Charles Darwin, 1809–1882

4.1 What computational models are

In Chapter 1 we have briefly introduced the neuroimaging approach to understand human cognition and its physical substrate, the brain. Such technique allows us to associate given behaviors with the activation of specific neural populations, so that, at least in principle, it is possible to identify sub-regions in the brain that are devoted, with some degree of specialization, to given processing tasks (vision vs. hearing, color processing vs. motion processing, etc.).

Neuroimaging experiments are complemented by behavioral experiments, both on healthy and pathological subjects: in these experiments, observed behaviors are associated with indexes such as error rates, reaction times, psychophysical thresholds, etc. From these measures, characteristics of the mental processes required to carry out the experimental task are inferred. For instance, a (statistically significant) higher error-rate in the pathological subject group than in the normal group suggests that the employed task recruits neural resources that are affected by the disease at hand; and lower reaction times for task 1 than for task 2 imply that task 1 is less demanding than task 2.

However informative, these approaches do not exhaust the spectrum of techniques that can be used to investigate the human mind. Besides these purely *experimental* approaches, we can also take advantage of *theoretical* approaches, or, more precisely, *computational* modelling. In computational modelling, the cognitive aspect of interest is interpreted in terms of processes carried out by a computational device, and implemented so that simulations can be

run, and that the data produced by the model can be compared to behavioral data collected from human subjects. It is important to remark here that the aim of cognitive modelling is that of advancing our understanding of how the mind works, rather than building computer systems that can execute a given task optimally and in the most efficient way possible. This might imply, for instance, that it is perfectly acceptable that our model makes mistakes in performing the task – because humans make mistakes too. The goal of cognitive modelling is not *engineering* a well-performing system (although ideas derived from cognitive models might guide the building of such system), but rather *reverse-engineering* the human mind in order to infer the general principles it rests on.

In this part of the Thesis, we will explore the potentialities of computational models of cognition. An overview of the history, motivations, and goals of computational modelling at large is given in this chapter, together with a detailed presentation of the connectionist approach. Chapter 5 focuses on a specific cognitive process, single word reading, providing a discussion on the existent computational models and related theoretical assumptions. This discussion prepares the ground for the introduction of our modelling work on reading, a connectionist network that conceptually distinguishes between a more cognitive component, responsible for the computation of an internal code for the input word, and an articulatory component that executes the phonological sequence corresponding to that code, thus introducing seriality in output production. A threshold-based mechanism is assumed to signal the articulatory portion of the network when a naming response can be initiated, based on the quality of the available internal code. The description of this model, together with a critical analysis of its value and weaknesses, constitutes Chapter 6.

4.1.1 History of cognitive science and modelling

Computational modelling[1] is as old as cognitive science and artificial intelligence, tracing back to the 1950-60s. These disciplines are in fact closely intertwined, and sometimes regarded as just the two sides of the same coin. Whereas cognitive science focuses on understanding how the mind produces cognitive faculties, artificial intelligence (AI) (Russell and Norvig, 1995) is more aimed at replicating such faculties in artificial devices. The very fathers of AI (John McCarthy, Marvin Minsky, Alan Newell, Herbert Simon) were actually deeply interested in the mechanisms of the mind; and most of them are also considered to be the first cognitive scientists.

An essential assumption in cognitive science is that the mind is a *computational device* (Churchland and Sejnowski, 1994); as such, its processes can be conceived as computations, as algorithms that return outputs in response to inputs. Such a position was influenced by the advancements in computability theory, first, and the rise of the digital computer, later. A key figure in this context was Alan Turing, who formalized the concept of computable func-

[1] For extensive treatment of the topic of computational modelling, in its various facets, the interested reader may find the following books useful: *The Computational Brain* (Churchland and Sejnowski, 1994); *Computational explorations in cognitive neuroscience* (O'Reilly and Munakata, 2000); *Theoretical Neuroscience* (Dayan and Abbott, 2001); *The Cambridge Handbook of Computational Psychology* (Sun, 2008a).

tion and introduced the idea of a computing machine (Turing, 1936), to some degree inspired by human computation ("*Computing is normally done by writing certain symbols on paper*", and that is how the Turing machine works – by writing on a tape). The view of computations as, essentially, manipulations of symbols according to some (transition) rules characterized the early age of AI and cognitive science, leading to GOFAI (good old-fashioned AI (Haugeland, 1985)) and *symbolic approaches* in cognitive science (Boden, 2008; Schonbein and Bechtel, 2003).

In symbolic approaches the objects of the computations of the mind are symbols, and computation itself is the application of rules for transforming such symbols. The physical symbol systems introduced by Newell and Simon, like the General Problem Solver (Newell and Simon, 1963), embodied this conviction. These systems were employed both to show that machines could actually be endowed with some form of intelligence (at least the one that is required to plan actions, or play chess), and to provide evidence that human cognition was based on symbolic rule application as well.

The approach soon gained popularity, but at the same time a different current was coming to life, *connectionism* (also referred to as a *sub-symbolic approach*). The origins of connectionism[2] can be traced back to the model of artificial neuron proposed by McCulloch and Pitts (McCulloch and Pitts, 1943). The model was inspired by basic features of the biological neuron, but was introduced as a powerful computational device rather than a truthful model of the neuronal cell. Interest in artificial neurons increased when Donald Hebb stated his rule for neural learning (Hebb, 1949). Now the artificial neuron could not only compute, but also *learn to compute*. When Rosenblatt introduced the perceptron, the first artificial neuron network endowed with a learning rule (Rosenblatt, 1958, 1962), connectionism was officially born.

However, in the late 60s both approaches found fierce critics. Symbolic approaches (GOFAI) were attacked by Hubert Dreyfus (Dreyfus, 1965), who challenged the assumption that human intelligence is based on manipulation of symbols, and rather favored the contribution of intuition and instincts, that are hardly formalized by rules. On the other hand, the spread of connectionism received a drastic halt when Minsky and Papert showed that a large class of problems (the non-linearly separable functions) cannot be computed by a perceptron (Minsky and Papert, 1969). The Dark Ages of connectionism opened, and it was only with the introduction of the backpropagation algorithm (Rumelhart et al., 1986) that connectionist approaches were rehabilitated, and actually started to gain increasing popularity.

Although symbolic approaches were still in good health, they were also targets for further critiques. One of the most notable came in the form of the Chinese Room argument (Searle, 1980). John Searle proposed this example to show that human intelligence cannot be achieved by pure application of symbolic rules: an English speaking person, with no knowledge of the Chinese language, is locked in a room. From outside the room, the person is given papers full of Chinese characters, and rules in English to map the received symbols into other Chinese characters that are returned to the outside world. The person in the room does not know

[2]Here, we intend to provide only a brief historical overview of connectionism; for technical details, see Section 4.2.1.

that he received questions in Chinese, and does not know what he replied to such questions; thus, there is no understanding (no real human intelligence), although his behavior, observed from outside the room, looks perfectly intelligent. More precisely, Searle's attack was targeted to the so-called "strong AI" position[3], according to which, if a computer could be built that acts intelligently, then it would also *be* intelligent – it would be a mind by itself (and as such also a good model for the human mind). According to Searle, Strong AI should be rejected *"[B]ecause the formal symbol manipulations by themselves don't have any intentionality"* (Searle, 1980, p. 422).

Disputes between the two main original currents characterized the whole history of cognitive science. Both currents are still lively nowadays, while new approaches are becoming increasingly popular (for instance, the Bayesian approach (Griffiths et al., 2008)). It is crucial to remark that these currents do not differ only in the kind of computational tools that are used to physically implement cognitive theories – the elected paradigm is assumed to be a model (in the broad sense), a metaphor of the actual cognitive processes.

4.1.2 Why modelling?

Computational models of various cognitive processes (may they be connectionist, or symbolic, or Bayesian) have been proposed all along the history of cognitive science. A full overview of the modelling effort over the last 50 years is not viable here, and probably not even advisable. Here, it is more interesting to try and answer a different question: rather than "what has been done in the field of computational cognitive modelling?", we wish to ask "*why* do we even want to embark in computational cognitive modelling?".

Let us start by saying that the computational assumption that characterizes cognitive science naturally leads to computational modelling as an important method of investigation: if our mind is, in fact, a computational device, we may gain knowledge about it by using another, to some extent similar, computational device. If we interpret a cognitive process in terms of computational steps, it is at least tempting to try and implement them on a computer.

More generally speaking, modelling is a powerful tool for describing and explaining phenomena in all sciences. It allows, at the same time, for rigorous formalizations and helpful simplifications of the problem at hand. Various types of modelling can be distinguished: verbal, mathematical, and computational (Sun, 2008b). Verbal, or box-and-arrow, models typically represents relationships between concepts, for instance sub-processes of a cognitive process, in a similar way as E-R models do for databases. They represent the translation of a verbal description into a simplified schema; most notably, they usually do not incorporate fine details about the postulated entities and their relations. Mathematical models are rigorous descriptions of the studied process, but are mostly descriptive, rather than explanatory (Dayan, 1994). More explanatory power is provided by computational models that, by implementing theoretical assumptions in fully-specified algorithms, offer a way to investigate *how*

[3] A softer claim, called "weak AI", was that machines acting in a seemingly intelligent way could possibly be built by means of appropriate programs.

a certain computation, possibly described by a mathematical model, can be realized by the brain.

The most powerful feature of computational modelling, with respect to other modelling approaches as well as experimental techniques, lies in that full specification of details of the considered cognitive theory must be necessarily given. Modelling a cognitive process – that is, coding an algorithm for it – forces the investigator to ponder the impact of all implementational choices, and to answer to specific questions that may be easily overlooked at when just providing a verbal account of an intuitive working hypothesis. What exactly is the input for this process? How the output of one sub-process interacts with the operations of another one? Which range of values is reasonable for a certain parameter, and what kind of psychological entity does that parameter stand for? These are just few examples of questions that must be answered before a model can be fully implemented. In this respect, computational modelling also offers an opportunity to explore alternative scenarios, corresponding to differently implemented details of the same cognitive theory.

Once a model has been implemented, simulations can be run in order to compare data from experimental subjects with the output of the model. A good fit of the data provides support to the plausibility of the implemented cognitive theory: it is, at least, possible, that the assumptions on which the model is based are compatible with the actual brain implementation. However, a definite proof of the cognitive theory at hand cannot be provided on this basis only; after all, *"behavior is compatible with a huge number of different computational hypothesis, only one of which may be true of the brain"* (Churchland and Sejnowski, 1994, chap. 2). Additional support (or, conversely, definite refutation) must be gained through further experiments; these, in turns, can take advantage of the model itself. In fact, a computational model typically generates predictions, that can be tested through ad-hoc experiments. In this sense, models represent a guide for experimenters, and help refine intuitive working hypotheses (Sun, 2008b).

On the other hand, a bad fit of the data generated by the model to the human ones does not necessarily mean that the implemented assumptions are thoroughly incompatible with what actually happens in the brain (McClelland, 2009). It is possible that the negative result is ascribable to few implementational details, like the choice of representations for the input. In this case, while the main architecture of the model (and therefore the main assumptions it is based on) may well be valid, unfortunate choices for some (possibly minor) aspect of the implementation may shadow the worthiness of the model as a whole. For example, the PMSP model (Plaut et al., 1996) drastically improved its predecessor, the SM model (Seidenberg and McClelland, 1989), by adopting different orthographic units as input to the model that, otherwise, remained basically the same (see Chapter 5 for further details).

Finally, it is worth remarking here that, although models are assumed to describe fully-functional cognitive processes, they also provide insights into disrupted cognition. In a similar way as a disease or an accident can damage the brain, resulting in some form of cognitive deficit, a well-designed model should be apt to be lesioned as well. The behavior of the

model once a part of it has been removed, or perturbed in some way, can then be compared to behavioral data from classes of patients. This process has the potential of both providing an explanation for specific neuropshychological deficits, and confirming the validity of the intact model to represent normal cognition.

4.1.3 Levels in modelling

As mentioned in Chapter 1, the study of the mind can be tackled at different levels. Different levels correspond to different questions being asked, different details being focused on, and different entities being investigated. Such level-oriented approach is applied to cognitive science in general, but also to cognitive modelling specifically.

Different level-based frameworks of analysis have actually been proposed. We have already cited Marr's proposal (Marr, 1982) of a computational, algorithmic, and implementational level. A similar hierarchy was advanced by Newell and Simon (1976), employing three levels for knowledge (including the agent's goals), symbols (how knowledge is implemented), and physical form (how symbols are realized) respectively. Both approaches are basically top-down: stress is given to the highest levels of the hierarchy, whereas the physical implementation is considered to be less relevant to the purpose of understanding cognitive processing. Alternative frameworks of level-based analysis take into account the different scales of anatomical organization in the brain (levels of organization), or the location of the considered process in an ideal stream starting from the peripheral collection of sensory input, and ending in the "higher cognitive functions" taking place in the cortex (levels of processing) (Churchland and Sejnowski, 1994). Another approach is proposed by (Sun et al., 2005), where levels of analysis and modelling are shaped on the range of phenomena addressed by the different disciplines in cognitive science: the sociological level (studying processes that involve multiple agents), the psychological level (studying the individual *per se*), the componential level (studying one, or few, modules of cognition), and the physiological level (studying the biological substrate of cognition). All these frameworks are useful in guiding, and constraining, the modelling work, by identifying the kind of questions that a model should address, and therefore helping sort out the relevant details from what can be, at least for the moment, overlooked.

Computational cognitive modelling traditionally tends to privilege a top-down approach: the focus is on reproducing the kind of behavior observed in humans without addressing in detail the question of how that behavior might be implemented in neuronal dynamics. However, although it is legitimate and even mandatory (for the sake of simplification) to work at one, maybe two, levels of analysis, other levels should not be completely disregarded, as, using the words of (Churchland and Sejnowski, 1994, chap. 1), "*research at one level provides correction, constraints, and inspiration for research at higher and at lower levels*".

Useful information, in this respect, can be provided by work in computational neuroscience, that typically focuses its modelling effort on the lowest level of analysis (this is true for most of the frameworks of analysis mentioned above). The objects of modelling, here, are

biological neurons and their interactions. Computational neuroscience has a stronger commitment to biological plausibility, although there is variability within this discipline as well: models at different levels of abstraction, and therefore more or less faithful to the biological properties of neurons, can be employed (conductance-based models, integrate-and-fire-models, and firing-rate models, in increasing order of abstraction) (Dayan, 2003). The focus of this discipline on neuronal functioning makes it mostly committed to a bottom-up approach: as cognition is ultimately the product of neurons firing in our brain, understanding how a firing pattern turns into an observable behavior is the main goal.

However, no purely top-down or bottom-up approach can, alone, explain how we think, act, and feel. For this reason, the ultimate, fully-explanatory model of human cognition, if such model is ever built, will have to integrate knowledge gained at each different level in a seamless way.

4.2 Connectionism

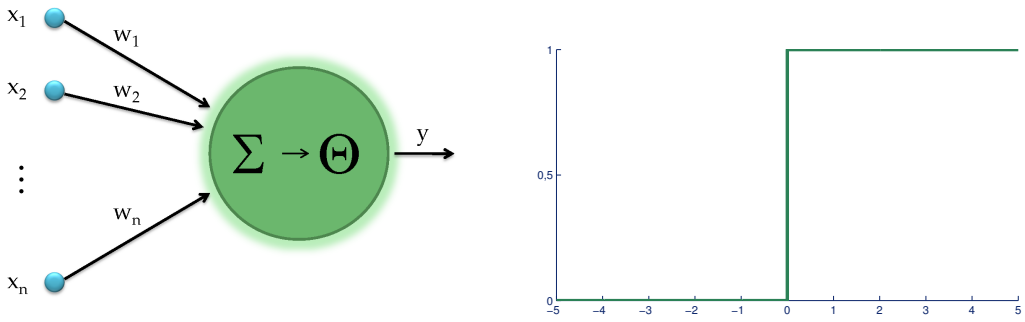
As we have mentioned in the previous section, different approaches to cognitive modelling have been proposed, and successfully applied to a range of tasks, along the history of cognitive science. As connectionism is the framework for our modelling effort, in what follows we present a review of this approach. A good introduction to other modelling paradigms can be found in (Sun, 2008a).

Connectionism, also referred to as the study of artificial neural networks, is a computational paradigm based on a set of interconnected, rather basic processing units, that *learn* to compute a specific function (input-output mapping) by storing relevant knowledge in their *connections* – hence the name. Artificial neural networks (ANNs – (Hertz et al., 1991; Haykin, 1999) have been extensively employed both as convenient tools in pattern recognition tasks (e.g. recognition of hand-written digits (Le Cun et al., 1989)), and as models of cognitive faculties; here, we will concentrate more on the latter domain, although the technical discussion is left unchanged by such observation. Notational conventions adopted through this chapter are described in the Appendix (A.4).

4.2.1 Artificial Neural Networks: technical overview

Artificial neurons, and networks based on them, have been introduced as formal logical devices inspired by principles of functioning of the biological neurons (McCulloch and Pitts, 1943). In particular, the McCulloch-Pitts neuron embodies the principle of all-or-none activity (that is, a neuron either fires or it does not, it cannot just partially fire[4]), which, in the two-valued propositional logic interpretation, may be seen as truth values; and it embodies the principle of integration of signals from incoming synapses as the mechanism responsible for such activity, provided that the neuron threshold is overcome. By this principle, different

[4]This is true of the amplitude of the single spike, of course; in neuronal coding, however, there do exist degrees in firing activity, that refer to the frequency within a spike train.



(a) (b)

Figure 4.1: Illustration of the McCulloch-Pitt model. *n* input terminals send data to the artificial neuron through weighted connections; the neuron computes the weighted sum of such inputs and pass it through a threshold (Panel (a)). In Panel (b), the activation function used by the unit: the step function.

neuronal units can be combined to form a network; and each network corresponds to a logical expression. The idea of a neural network as a computational device was born.

From then on, many different types of networks have been proposed. These can differ along various dimensions:

- the kind of units they employ (e.g. binary vs. continuous-valued) and the activation function they compute (e.g. sigmoid vs. gaussian);
- the architecture of the network, that is the way its units are arranged and connected (e.g. two-layer vs. multi-layer network, feed-forward vs. recurrent);
- the learning paradigm, that is how the environment interacts with the network to have it learn a task (e.g. supervised vs. unsupervised learning, batch vs. on-line learning);
- the learning rule, that is how the network adapts itself for executing the target task based on the provided information (e.g. backpropagation vs. competitive learning).

Many variants are thus possible, corresponding to different combinations of choices for the above aspects, although these are not completely independent: for instance, backpropagation cannot be used in combination with binary threshold unit (as their activation function is not differentiable). In what follows, we will illustrate some of the most popular network types that have characterized the history of the ANN field.

Binary threshold units: McCulloch-Pitts model

Possibly the simplest neural network is represented by a single McCulloch-Pitts unit (Fig. 4.1(a)). This is a binary threshold unit that receives inputs (from sensors, for instance) through

incoming connections; each connection has a weight that reflects its strength, the degree of its contribution to the activation of the unit. The weighted sum of the inputs at the current time is called *net input* of the unit. The unit maps its net input into an activation value (its output) through the so-called *activation function*. A threshold unit also accommodates a value ϑ, also known as bias, that needs to be exceeded by the net input for the unit to be activated. Thus, the activation value $y(t)$ of the unit at time t is given by

$$y(t) = f\left(\sum_{j=1}^{n} w_j x_j(t-1) - \vartheta\right) \tag{4.1}$$

where each $x_j(t-1)$ is the value at time $t-1$ of the j-th input to the unit, and w_j the weight of the corresponding connection. In the case of the McCulloch-Pitts unit, the activation function $f(.)$ is the Heaviside (or step) function (Fig. 4.1(b)):

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

Therefore, the unit will be on only if the net input is larger than the threshold for activation. As mentioned, one such unit is able to represent a (simple) logical proposition: for instance, it is easy to see that, in the case of two binary inputs, if $w_1 = w_2 = 1$ and $\vartheta = 2$, then the unit is computing the binary AND function. Also notice that many other choices for these three parameters would accomplish the same goal.

Hebbian learning

Assembling various neurons into a network increases the available computational power. However, at the early stages of connectionism the ability of compute any function had to be hand-coded into the connection weights. No learning procedure had been devised yet. An impulse for this was provided by Donald Hebb, whose statement on how synaptic plasticity might take place in the brain (Hebb, 1949):

> *When an axon in cell A is near enough to excite cell B and repeatedly and persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency in firing B, is increased.*

laid also the foundations for learning in artificial neural networks (and most learning procedures are still called Hebbian to recognize his influence). Since knowledge is stored in the weights of the connections, learning must involve the modification of such weights. Hebb's rule suggests how they might be modified: by increasing the weights of connections linking two correlated units. Therefore, the basic Hebb's rule states that

$$\Delta w_{ij} = \eta y_i y_j \tag{4.3}$$

where w_{ij} is the weight of the connection from unit j to unit i, and η is a constant known as *learning rate*. Hebbian learning is also referred to as associative learning, because it discovers and strengthens associations – correlations – between activations of neurons.

However, there are at least two problems with this rule: first, it can only increase weights (as $y_i, y_j \in \{0, 1\}$), and therefore it eventually leads to infinitely growing weights; second, learning a task can be hard as, in typical unsupervised hebbian learning paradigms, there is no indication of what the network should learn. A solution to the first problem consists in letting the weights decrease as well, like in the BCM rule (Bienenstock et al., 1982), or ensuring that all weights are normalized, like in Oja's rule (Oja, 1982). A solution to the second problem consists in providing the network with expected outputs, or *targets*, to which actual outputs are compared. The introduction of targets in the learning mechanism characterizes *supervised learning*. In supervised learning, the training set is made up of input-target pairs, $TS = \{(\boldsymbol{x^1}, \boldsymbol{d^1}), \ldots, (\boldsymbol{x^l}, \boldsymbol{d^l})\}$[5], from which the network must extract the relevant statistical input-output relations in order to be able to successfully learn the assigned task. The performance of a network after training is measured in terms of its accuracy over the training set (percentage of training patterns for which the desired output is produced) and over a different test set, that contains input patterns to which the network has not been previously exposed, to assess its ability to generalize (see farther in this chapter for more comments on the issue of generalization). In contrast, if no feedback from the environment is given to the network, we talk of *unsupervised learning*; in this case, the network must discover regularities within the data set by itself, with no guidance on the desired outcome of the computation (as it is the case for clustering, see Chapter 2). On the middle ground we find *reinforcement learning* (Sutton and Barto, 1998): in this paradigm, only a qualitative feedback is given (right/wrong response), in place of the full target.

The perceptron

We have seen that Hebbian learning operates by reinforcing correlations between neuronal activities: it learns from observed associations. A different approach is that of learning from *errors*: each time an error is made (that is, what is produced by the network is not what it is expected from it), it is a sign that the network has not mastered its task and must be corrected. This is called *error-correcting* learning and is at the basis of the perceptron convergence algorithm (Rosenblatt, 1958). Let us consider a binary threshold unit like the one discussed above, with a slight modification: the activation function is now the sign function, that is

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{4.4}$$

[5]Until this point we treated the simplified case of a single unit, but in the most general case, inputs and desired/actual targets are better represented as vectors, specifying a value for each input and for each output unit – hence the bold notation.

Therefore, units have activation values in $\{-1, 1\}$ rather than in $\{0, 1\}$. Moreover, for the sake of simplicity, we omit the explicit declaration of the threshold ϑ, by introducing an additional, fictitious input x_0 being fixed at value -1, and correspondingly a weight $w_0 = \vartheta$ from this input to our unit. In this way, the contribution of the threshold is incorporated in the net input, and ϑ can be learned as any other weight.

The idea driving error-correcting learning is that of changing the weights whenever an error occurs (i.e., the desired output, d, is different from the actual output y), by increasing those for "good" connections and decreasing those for "bad" connections. Formally:

$$\Delta w_j = \begin{cases} 0 & \text{if } d \cdot y = 1, \text{ that is } d = y \\ 2\eta d x_j & \text{otherwise} \end{cases} \tag{4.5}$$

or, alternatively, $\Delta w_j = \eta(d - y)x_j$. It can be readily seen that the perceptron learning rule has an Hebbian inspiration. The rule can be applied also to the case when we have multiple output units forming a two-layer feed-forward network: that is, each input is connected to each output unit, and there are no further connections. It has been shown that this rule converges in finite time to a set of weights that implement the target function.

Linear units and the delta rule

A related learning rule is the so-called *delta rule*, also known as least-mean-square algorithm – LSM (Widrow and Hoff, 1960)). This rule has been derived for linear units, that is for units whose activation value is just the net input. The delta rule is a gradient descent algorithm in that it defines an error function and weights are modified in the direction of maximum decrease of this function. The error function is defined as

$$E(\boldsymbol{w}) = \frac{1}{2}\delta^2 \tag{4.6}$$

where

$$\delta = d - y = d - \sum_{j=0}^{n} w_j x_j \tag{4.7}$$

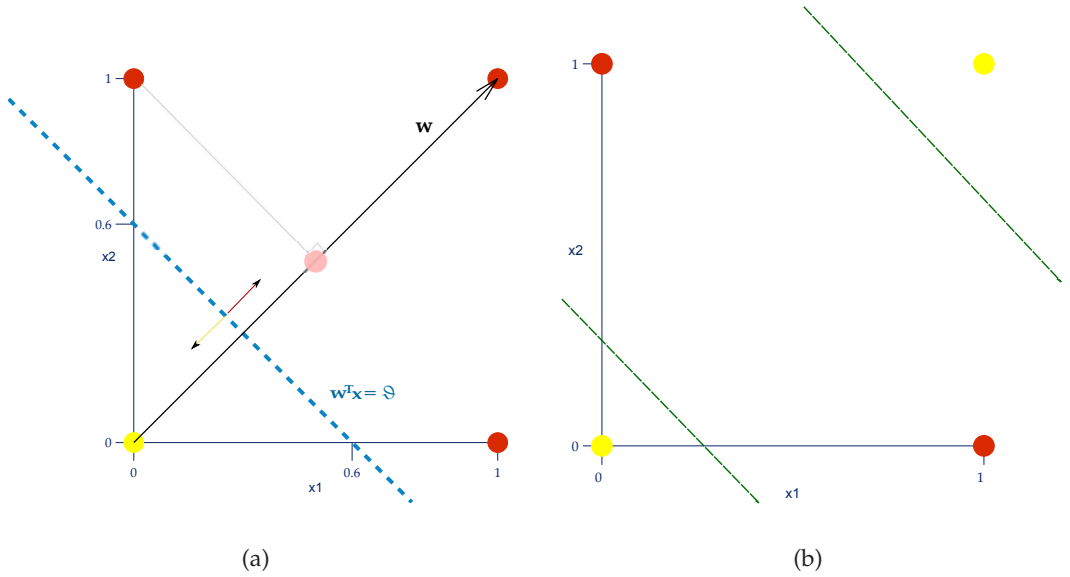
To find the direction of error decrease we have to differentiate $E(\boldsymbol{w})$ with respect to $\boldsymbol{w}$:

$$\frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} = -\delta \boldsymbol{x} \tag{4.8}$$

The delta rule changes weights by moving of a step η in the opposite direction of the gradient:

$$\Delta \boldsymbol{w} = \eta \delta \boldsymbol{x} \tag{4.9}$$

The delta rule has the same formulation as the perceptron learning rule (cf. Eq. 4.5), but while the latter was derived empirically for binary units, the former was obtained for linear units



(a) (b)

Figure 4.2: Linear separability (and non-separability) in a geometric interpretation. Panel (a) illustrates the OR problem: input patterns $[1,0]$, $[0,1]$, and $[1,1]$ corresponds to a desired output of 1, and are shown in red circles; pattern $[0,0]$ – in yellow – is associated with 0 output. The weight vector $w = [1,1]^T$ for a perceptron trained to learn this function is shown; the threshold is $\vartheta = 0.6$. These parameters define therefore a decision line having equation $x_1 + x_2 = 0.6$. The decision rule of the perceptron corresponds to projecting each input pattern onto the weight vector (see light-red circle for an example), and checking in which of the two subspaces determined by the decision line (that is, $x_1 + x_2 < 0.6$ or $x_1 + x_2 \geq 0.6$) such projection lies. Here it can be seen that the OR function was successfully learned; the OR problem is linearly separable. Conversely, the XOR problem is represented geometrically in panel (b): red circles indicate input patterns for which the desired output is 1, yellow circles those having target 0. No single decision line can be found to correctly separate the four input patterns into these two classes: the XOR problem is not linearly separable. Two lines or a curve are required as decision surfaces.

by performing gradient descent on the error surface.

Note that until now we have implicitly assumed that weight modifications are computed after a single training example has been presented: this is known as on-line learning modality. In batch approaches, on the other hand, typically a complete sweep through the training set is performed before weights are modified; in this case, errors on single trials are just accumulated (i.e., summed) and weights are then updated based on the accumulated error.

The shortcomings of two-layer feed-forward networks

In the 1960s, thus, the state of the art in ANNs included feed-forward networks with a single layer of output units, and learning rules inspired by Hebb's principle to train them. However, such networks had important limitations (Minsky and Papert, 1969). Most notably, they could compute only a sub-class of functions, the linearly separable functions. The classical example of a function, albeit very simple, that cannot be computed by a perceptron is the log-

ical exclusive OR (XOR), which returns 1 if either of its arguments is 1, but not both of them are. To see why this goes beyond the capabilities of a perceptron, it is convenient to resort to a geometrical interpretation (Fig. 4.2). Let us recall that the output of a perceptron is 1 if the net input is over threshold, 0 otherwise. Note that this a typical *classification* problem: each input must be assigned to either of two categories. The vector of the weights in the perceptron, $\boldsymbol{w}$, can be visualized in the input space (see Fig. 4.2(a)). The inner product computing the net input, $\boldsymbol{w}^T\boldsymbol{x}$, corresponds to a projection of the current input pattern onto this line. Let us now consider a line perpendicular to the weight vector, and intersecting x_2 axis at ϑ[6]. This is called *decision hyperplane* and has equation $\boldsymbol{w}^T\boldsymbol{x} = \vartheta$; if the input space is two-dimensional, as it is the case for this example, $\boldsymbol{w}^T\boldsymbol{x} = \vartheta$ simply defines a line. The thresholding step corresponds to assigning the output 1 to all input patterns whose projection lies to the right of the decision hyperplane, and 0 to the remaining ones. For the XOR problem, it can be readily seen that no single line is able to assign the right output to all possible input patterns (Fig. 4.2(b)); a decision surface consisting of two lines, or a curve, would be needed, but this cannot be realized with a perceptron. The perceptron is, in fact, a linear associator, and as such it can only solve linearly separable problems: that is, problems whose input patterns can be separated in the correct classes by means of a linear hyperplane.

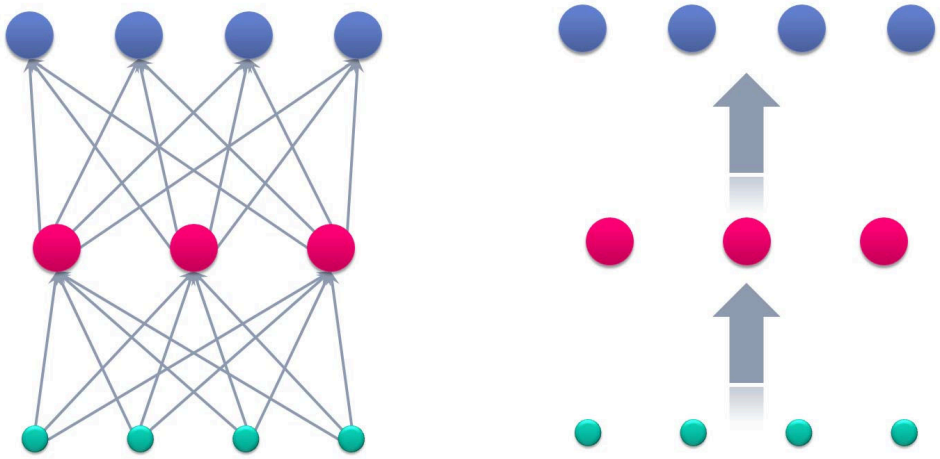
Multi-layered feed-forward networks and the backpropagation algorithm

However, it was known that multi-layered networks with nonlinear units, that is networks having one or more intervening layers of units with a nonlinear activation function before the output layer (see Figure 4.3), were more computationally powerful. In fact, it was proved (Cybenko, 1989; Hornik et al., 1989) that a feed-forward multi-layer network can approximate any continuous differentiable function, provided it has a large enough number of nonlinear units. The problem was that no learning rule for such networks existed yet. The delta rule and its variants are error-correcting algorithms that change weights based on the discrepancy between the actual output and the desired target; but no target could be easily provided for intermediate, non-output units (these are called *hidden units*, because they are not directly visible by the environment), and thus no error could be defined on them.

The invention of a rule for training multi-layer networks was therefore an important development in the history of ANNs. Such rule was called *backpropagation*[7], because it involves propagating backward to all units a signal error computed at the output layer; different researchers were independently responsible for its formulation (LeCun, 1985; Parker, 1982; Rumelhart et al., 1986). The backpropagation algorithm essentially generalizes the delta rule, in that it defines a measure of global error and performs gradient descent on it – it computes its derivative. This implies that the activation function of the units in the network must be differentiable (that is, no step function can be used). A commonly employed activation function

[6] More precisely, the decision hyperplane intersects the weight vector at a distance $\vartheta / \|\boldsymbol{w}\|$ from the origin.

[7] A learning rule for Boltzmann Machines – see farther in this section – was also proposed in the same years (Ackley et al., 1985), which shortly preceded the introduction of the backpropagation algorithm.



(a) (b)

Figure 4.3: A typical feed-forward network architecture. The green circles represents the input to the network; the magenta ones represent the hidden units, and blue ones the output units. Connections are only from one layer to the next, and are typically all-to-all. The right panel shows a more compact representation where the all-to-all connections between two layers are depicted by one arrow only.

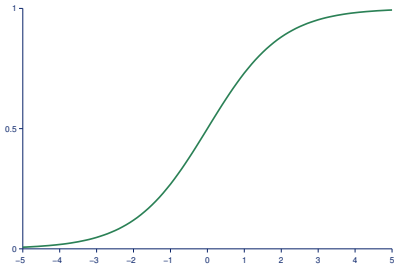


Figure 4.4: The logistic activation function.

is the logistic function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.10}$$

that is basically a smooth version of the step function (see Fig. 4.4). This function is also convenient in that it has a particularly simple derivative: $f'(x) = f(x)(1 - f(x))$. Notice that in this case we working on continuous-valued units, rather than binary ones, whose activation values are bounded in the $(0, 1)$ interval.

Backpropagation works through a forward and a backward step. During the forward pass, activation is propagated onward, layer after layer, to the m output units. Here, the

global error of the network,

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{m}(d_m - y_m)^2 \tag{4.11}$$

can be computed. We are assuming here an on-line training mode, so the error depends only on the current training item. In the backward pass, the error signal – the derivative of the error with respect to the weights – is propagated back so that blame and credit for the network performance can be assigned to single hidden units. The error-correcting rule in backpropagation assumes the general form:

$$\Delta w_{pq} = \eta \delta_p y_q \tag{4.12}$$

where δ_p is the signal error computed for unit p. The way it is defined depends on which set of connections we are updating. If we are working on the hidden-to-output connections, then

$$\Delta w_{ip} = -\eta \frac{\partial E}{\partial w_{ip}} = \eta(d_i - y_i)f'(net_i)y_p = \eta \delta_i y_p \tag{4.13}$$

where net_i is the net input to unit i. Note that, unsurprisingly, this is exactly the delta rule introduced above (Eq. 4.9).

For connections from the input to the hidden layer, we can then use the chain rule for derivatives:

$$\Delta w_{pj} = -\eta \frac{\partial E}{\partial w_{pj}} = -\eta \frac{\partial E}{\partial y_p}\frac{\partial y_p}{\partial w_{pj}} = \eta \left(\sum_{i=1}^{m}(d_i - y_i)f'(net_i)w_{ip}\right)\left(f'(net_p)x_j\right) = \eta \delta_p x_j \tag{4.14}$$

where this time the error signal δ_p is defined as $\delta_p = f'(net_p)\sum_{i=1}^{m}(d_i - y_i)f'(net_i)w_{ip}$ or, more succinctly, $\delta_p = f'(net_p)\sum_{i=1}^{m} w_{ij}\delta_i$. So the error signals, our deltas, are backpropagated layer after layer to drive the weights update. If more than one layer of hidden units is present, the same computations can be performed to derive the appropriate rule.

The backpropagation algorithm and its variants are still widely used nowadays, although some drawbacks exist in their application (Frean, 2003; Munro, 2003). Like every gradient descent approach, backpropagation can sometimes get trapped in local minima of the error surface. This is particularly true if a large value of the learning rate η is used; on the other hand, small ηs can make convergence rather slow. A partial remedy to this problem consists in using an adaptive learning rate that is decreased when oscillations are detected, or increased when we want to speed up the learning process because we are confident that we are moving in the right direction. For instance, in the delta-bar-delta approach (Jacobs, 1988), a separate learning rate η_{pq} is used for every weight, and updated so that if the current error gradient with respect to that connection is in the same direction as previous gradients, η_{pq} is increased, otherwise it is decreased. Another approach is that of adding a *momentum* term $\alpha \in [0, 1]$

to the weight update rule (Plaut et al., 1986) that promotes changes in the weights that are consistent with previous updates (and therefore punishes oscillations):

$$\Delta w_{pq}(t+1) = -\eta \frac{\partial E}{\partial w_{pq}} + \alpha \Delta w_{pq}(t) \tag{4.15}$$

This additional constraint allows for larger values of η to be used.

The derivation of the backpropagation algorithm starts by defining an error function to be minimized. We have introduced a definition of error function – the squared error, see Eq. 4.11 – that is very natural, but it is not the only possible one. In fact, when used in combination with a logistic activation function, this error function can be responsible for slow learning. To see why, let us consider an output unit which should be on but, instead, is off (or the reverse). In other words, $d_i = 1$, $y_i \approx 0$ ($d_i = 0, y_i \approx 1$). Let us consider the update rule in Eq. 4.13: if $f(.)$ is the logistic function, then $f'(net_i) = f(net_i)(1 - f(net_i)) = y_i(1 - y_i)$. The update rule for weight w_{ip} can then be rewritten as:

$$\Delta w_{ip} = \eta(d_i - y_i)y_i(1 - y_i)y_p \tag{4.16}$$

It is easy to see that, in the case we are considering, even though the actual error $d_i - y_i$ is large (close to ± 1), the term $y_i(1 - y_i)$ will be close to 0, thus making the weight update very small – and in a situation where we would like to have a strong modification of weights, since we are registering a large error. This problem can be avoided by using a different error function, namely the *cross-entropy* (Hinton, 1989):

$$C = -\sum_{i=1}^{m} d_i \log_2(y_i) + (1 - d_i) \log_2(1 - y_i) \tag{4.17}$$

The update rule corresponding to the minimization of this function can be derived as follows:

$$\frac{\partial C}{\partial w_{ip}} = -\left(d_i \frac{\log_2 e}{y_i} f'(net_i) y_p - (1 - d_i) \frac{\log_2 e}{1 - y_i} f'(net_i) y_p \right) \tag{4.18}$$

$$\Delta w_{ip} = -\eta \frac{\partial C}{\partial w_{ip}} = \eta(\log_2 e) \left(\frac{d_i y_p f'(net_i)(1 - y_i) - (1 - d_i) y_p f'(net_i) y_i}{y_i(1 - y_i)} \right) \tag{4.19}$$

If we are using a logistic activation function, it follows that:

$$\begin{aligned} \Delta w_{ip} &= \eta(\log_2 e) \left(\frac{d_i y_p y_i (1 - y_i)^2 - (1 - d_i) y_p y_i^2 (1 - y_i)}{y_1(1 - y_i)} \right) = \\ &= \eta(\log_2 e) \left(d_i y_p (1 - y_i) - (1 - d_i) y_p y_i \right) = \\ &= \eta(\log_2 e) \left(y_p \left(d_i - d_i y_i - y_i + d_i y_i \right) \right) = \\ &= \eta(\log_2 e)\, y_p (d_i - y_i) \end{aligned} \tag{4.20}$$

Note that this rule is (neglecting the constant term) like Eq. 4.13 but without the activation

function derivative term. As a result, the amount of weight update depends only on the magnitude of the error, and it is not shrunk by output units having an activation value close to the extremes.

Another point of concern in using multi-layer feed-forward networks lies in the choice of the architecture. There is no clear and warranted recipe for choosing the number of hidden layers[8], and the number of units in each layer. Few units can hamper the ability of the network to learn its task; on the other hand, a too large network is slower to train, and more prone to the risk of *overfitting*. In other words, the network may be able to perfectly master the task as far as the training set is concerned, but lacks in generalization – it cannot perform in a satisfactory way when tested on novel items. When a network overfits, it is basically doing curve fitting (including noise fitting), but it is not learning to extract any statistical regularity from the training data to be successfully applied to the testing set as well. To limit the problem of overfitting, one possibility is to stop training the network when the training error is acceptable and, at the same time, the testing error stop decreasing. *Weight decay* (Hinton, 1986) can also help: in its basic form, it consists in subtracting from each weight a fixed proportion of it after each update. This corresponds to adding to the error function a penalty term that is proportional to the squared sum of the weights: large weights are therefore penalized. As large weights amplify noise in the data, weight decay promotes generalization and stability of the network. On the other hand, it can slow down the training process, and for this reason the weight decay parameter should be very small.

As overfitting depends on the complexity of the network, an alternative approach to control it consists in having the network discover by itself the optimal architecture for the task at hand, through adaptive processes of growth and pruning. That is, one can start with a large network and progressively eliminate the less important units, or weights, based on some measure of saliency; for instance, in (LeCun et al., 1990) the saliency of a parameter is defined to be proportional to the Hessian of the error function for that parameter, and used to discard one or more parameters from the current network; the reduced network is then re-trained, and the next pruning step begins. This process is continued until the generalization error starts increasing. Or, the reverse could be done, starting with a minimal network and adding units until the error of the network becomes satisfactory. Two such examples are the upstart algorithm (Frean, 1990), for binary units, and the cascade-correlation algorithm (Fahlman and Lebiere, 1990), for real-valued units: in both algorithms, units are added to correct for mistakes made by previously introduced units, thus forming a hierarchy. It should however be remarked that, for large-scale problems, the most common practice is to try different sensible combinations of network parameters (this often includes introducing moderate weight decay), and choose the best performing one.

Notwithstanding the above limitations – rate and quality of convergence, overfitting risk, difficulties in choosing the appropriate architecture – feed-forward network have been, and still are, widely used for pattern recognition and regression problems. The key of the com-

[8]Note, however, that two hidden layers have been proved to be sufficient for approximating any function (Cybenko, 1988).

putational power of multi-layer networks is, of course, in the hidden units. Their role is that of re-mapping the information carried by the input to an internal representation that is most useful for achieving the desired computation. This internal representation draws on the statistical information embodied in the data, but it is usually not transparent to the outside. Therefore, interpreting what exactly a hidden unit is representing can be hard. Incidentally, this is one of the classical criticisms of connectionist models of cognition: their sometimes hard to uncover internal operations would make them of little use for advancing our insight into the processes they are supposed to model. Another criticism involves the biological implausibility of the backpropagation algorithm itself. We will come back to these issues in the next section.

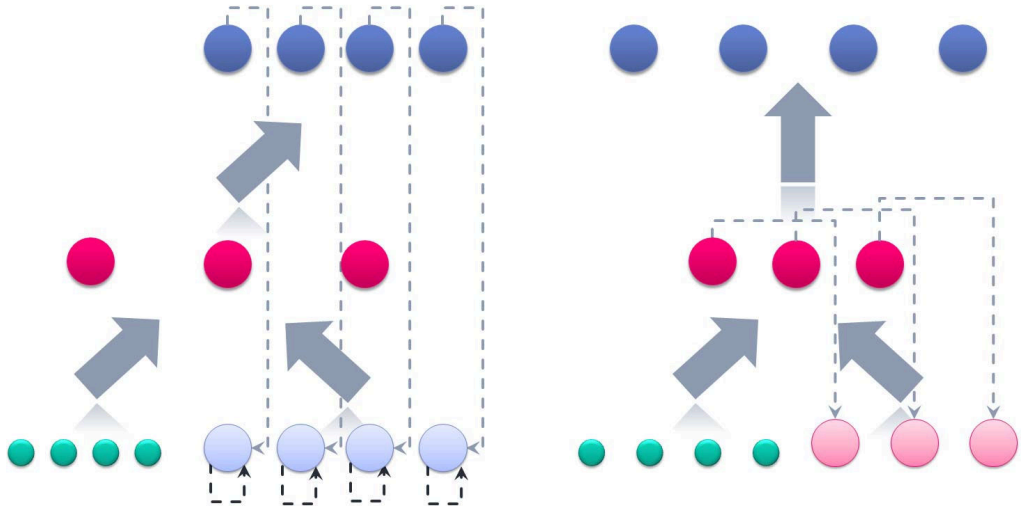
The temporal dimension: recurrent networks

In feed-forward neural networks, as we have seen, only connections from units in one layer to units in the subsequent layer are allowed. However, it is natural to think of backward (and lateral, among units in the same layer) connections, whose contribution would be that of providing some form of feedback to previous layers. The idea of building *recurrent* network was also inspired by what it is known about brain organization: information does not flow mono-directionally, and feedback connections are as dense as feed-forward ones. The main feature of recurrent networks lies in that they can deal with the temporal dimension of data: adding feedback connections basically corresponds to providing the network with memory – which allows it to process sequential data.

A first attempt to add a temporal dimension to neural networks was done with *time-delay neural networks*, that are actually still purely feed-forward networks. They manage to introduce temporal dynamics by turning a temporal sequence into a *spatial* sequence: temporal relations between events (before/after) are represented as spatial relations (left/right). The network is endowed with an input window of fixed size s that accommodates the currently available subsequence, of length s, of the input. If the temporal sequence that must be passed to the network is $(x(0), x(1), \ldots, x(z))$, then at each step t the actual input is the sequence $(x(t), x(t-1), \ldots, x(t-s))$. An example of such kind of network is NETtalk (Sejonwski and Rosenberg, 1986; Sejnowski and Rosenberg, 1987), to which we will return in the next chapter. Time-delay networks can perform sequence recognition tasks, but are limited in that the size of the input window must be fixed in advance, and can require a large number of input units (for instance, if each component of the temporal sequence can assume one of k values, coded by having one input unit on and the other $k-1$ ones off, we need sk input units in total) which in turns impacts training time.

To obtain more flexible approaches to dealing with time, it is required to abandon purely feed-forward architectures. Different network typologies have been proposed that vary on how feedback is implemented and propagated: the main distinction in this sense is that between simple recurrent networks (SRNs) and fully recurrent ones.

Jordan networks (Jordan, 1986) and *Elman networks* (Elman, 1990) are classical examples of



(a) The architecture of a Jordan network.

(b) The architecture of an Elman network.

Figure 4.5: Context units in these networks are represented in lighter color than the layer from which they are copying activations. The dashed arrows represent one-to-one connections with frozen weights. The gray arrows stand for connections having weight equal to 1; the darker arrows represent connections having weight α.

SRNs. These networks are characterized by the presence of a so-called *context* layer that hosts copies of previous activation values of units in another layer. In a Jordan network (see Figure 4.5(a)), the context layer keeps a trace of the previous activation value of output units; in other words, there are feedback one-to-one connections from the output layer to the context layer with frozen (i.e., unmodifiable) weights equal to 1. These connections therefore implement the copying process of activation values. The context layer also feeds into itself with frozen weights α, so $y_C(t+1) = \alpha y_C(t) + y_O(t)$, with y_C denoting the activation value of a given context unit, and y_O denoting the activation value of the corresponding output unit. Both the context layer and the input layer feed into the hidden layer: at each processing step, then, hidden units see the current input and a decaying trace of previous outputs. Here it becomes clear what we mean when we say that recurrence endows networks with memory. In an Elman network (see Figure 4.5(b)), the context layer stores copies of hidden layer activations instead, so that the current state of a hidden unit depends on the current input and the internal representation of the previous input. In both cases, as the feedback connections are frozen, there is no need to introduce a new training algorithm, and backpropagation can be employed as it is in feed-forward networks. In general, unit updates in SRNs occur like they do in feed-forward networks, that is, once per layer, in a sequential fashion (the input layer is updated first, followed by the hidden layer, and so on).

Conversely, updates in *fully recurrent networks* are usually computed in a synchronous manner, and for multiple time steps. In these networks, the activation of a unit (and therefore

its contribution to the network error) can depend on the activation at the previous step of any other unit in the network, including itself:

$$y_p(t+1) = f(net_p(t+1)) = f\left(\sum_q w_{pq} y_q(t)\right) \tag{4.21}$$

with p, q any two units in the network.

This complicates the process of assigning credit/blame to connections to modify weights. Backpropagation cannot be directly applied here; however, variants of it have been developed to be used in recurrent networks. One popular variant is called *backpropagation through-time* – BPTT (Rumelhart et al., 1986). This algorithm is based on the idea of unfolding the original recurrent network into a feed-forward one: if we are considering input/output sequences of maximum length T, then each unit in the original network is replicated T times in the unfolded one, and each copy represents the state of the original unit at a given time step. In a way, this is similar to the principle of time-delay networks: temporal dynamics are turned into spatial relations. Connections in the unfolded network are inserted so as to respect the original architecture: so, for instance, if unit q fed into unit p and into itself, then copy q_t in the feed-forward network would feed into units p_{t+1} and q_{t+1}. The resulting feed-forward network can be trained by the usual backpropagation algorithm; however, since all connections between units q_t and p_{t+1}, for each t, in the unfolded network actually represent the same connection in the original network, having (one) weight w_{pq}, care must be taken so that the learning rule does not modify differentially the corresponding weights. This is done by accumulating the updates for the same connection, and finally modifying the corresponding weights by the same total value.

Another approach to the training of recurrent networks is *real-time recurrent learning* (Williams and Zipser, 1989). As in standard backpropagation, an error measure is defined

$$E = \sum_{t=1}^{T} E(t) = \sum_{t=1}^{T} \frac{1}{2} \sum_k (d_k(t) - y_k(t))^2 \tag{4.22}$$

and its gradient computed. We denote with $E_k(t) = d_k(t) - y_k(t)$ the error for unit k at time t; if no target is defined for unit k at time t, the corresponding error $E_k(t)$ is just set to 0. The resulting rule, then, is:

$$\Delta w_{pq} = \sum_{t=1}^{T} \Delta w_{pq}(t) \tag{4.23}$$

$$\Delta w_{pq}(t) = \eta \sum_k E_k(t) \frac{\partial y_k(t)}{\partial w_{pq}} \tag{4.24}$$

$$\frac{\partial y_k(t)}{\partial w_{pq}} = f'(net_k(t-1)) \left(\bar{\delta}_{kp} y_q(t-1) + \sum_l w_{kl} \frac{\partial y_l(t-1)}{\partial w_{pq}}\right) \tag{4.25}$$

where we have indicated with $\bar{\delta}$ the Kronecker delta function[9]. From these equations it can be seen that the derivative of y_k for any unit k at time t depends recursively on past derivatives. Then, by starting from the initial condition $\frac{\partial y_k(0)}{\partial w_{pq}} = 0$ for each unit k, weight updates can be computed from the above formulas. The algorithm is called real-time because it is also allows for weight modifications to occur at every step, rather than after the whole sequence has been processed. The real-time mode is, in fact, more efficient, because it does not require the storage of partial weight updates. Nonetheless, it still requires to store derivatives from the previous step, for a space complexity of $O(N^3)$ for a fully connected network of N units. Time complexity is even higher ($O(N^4)$), because each derivative require approximately $O(N)$ time to be computed. Several variants of both these approaches (BPTT and RTRL) have been proposed (see Williams and Zipser, 1995), with a particular concern on reducing the computational cost of training a recurrent network.

Recurrent networks model dynamical systems, that is, systems characterized by a state that changes in time. The system state can be thought of as a point that moves in time, thus defining a trajectory. Thus, what is learned by a recurrent network are actually trajectories through the state space. Stable states within the network trajectory are called *attractors*. An attractor is surrounded by other points that form its basin of attraction: patterns of activation that fall within this basin are "attracted" to that stable state.

During learning, a dynamical system can pass through bifurcations, that is discontinuity points where a small, smooth change in the network parameters determine an abrupt modification of the output of the network. If a gradient descent-based learning algorithm is being employed, this translates to a large error signal being given, resulting in instability of the network. It follows that convergence to a (local) minimum cannot be guaranteed (Doya, 1992). Gradient-descent methods are also not able to provide the network with a long-range memory, as error derivatives exponentially decrease with time (Bengio et al., 1994). For all these reasons, recurrent networks tend to be hard to train. A strategy that has often been found to help learning in recurrent networks is *teacher forcing*: activation values of output units are substituted with the corresponding targets at every interval for which a target is specified (after computing the error for those units). In this way, the actual error on the forced units is backpropagated to previous time intervals, but subsequent intervals are not affected (because after teacher forcing, no error is produced by those units). In other words, the network is trained to produce later targets on the assumption that earlier targets were produced correctly, and learning the whole task is reduced to learning how to transition from one target to the next one.

It is worth mentioning that some special types of recurrent networks, collectively known as *reservoir computing* (RC – including Echo State Networks (Jaeger, 2001) and Liquid State Machines (Maass et al., 2002); for a review, see (Lukoševičius et al., 2009)) can overcome some of the outlined problems as a result of a more constrained architecture. RC networks are basically divided in two parts: the reservoir, that is a recurrent network whose connections

[9]This is defined as $\bar{\delta}_{pq} = 1$ if $p = q$, 0 otherwise.

are randomly initialized and get no training; and a readout network, that receives trainable connections from the units in the reservoir, and determines the output by computing a linear combination of their activities. As the readout part of the RC network is strictly feed-forward, training it poses no particular problems. RC networks are reported to have excellent performances in benchmark tasks, and there is also support for their biological plausibility. The main downside lies in that the reservoir is randomly generated, and therefore there are no guarantees of optimality.

Finally, we introduce another feature in our neural networks: gradualness. Until this point we have always assumed that computation in a neural network proceeds in discrete steps: activation of a unit changes completely at each time step, depending on current inputs. If more gradual activations are desired, a continuous network can be used. Here, units integrate their output (also input integration is possible) over time, as prescribed by the differential equation:

$$\tau_p \frac{dy_p}{dt} = -y_p + f(net_p) \tag{4.26}$$

For implementational purposes, however, a discrete approximation can be used (Pearlmutter, 1989):

$$y_p(t + \Delta t) = \left(1 - \frac{\Delta t}{\tau_p}\right) y_p(t) + \frac{\Delta t}{\tau_p} f(net_p(t)) \tag{4.27}$$

This follows from the approximation of the derivative in (4.26):

$$\frac{dy_p}{dt}(t) = \frac{y_p(t + \Delta t) - y_p(t)}{\Delta t} \tag{4.28}$$

When $\Delta t / \tau_p = 1$, we go back to a fully discrete dynamics. Thus, the output of a unit changes gradually, with updates occurring with sampling interval Δt, from its current value to its asymptotic new value; each update step is sometimes referred to as a *tick*. Continuous versions of training algorithms for recurrent networks have been proposed (Pineda, 1987; Pearlmutter, 1989). The error function is defined as an integral of the error over time, and its minimization yields backpropagated error signals that are themselves integrated over time. Although the gradualness in the response of a continuous network can be a very desirable property, as it allows for integration of information over time and it is less sensitive to abrupt changes in the input, for these same reasons these networks can be hard to train. In fact, even though the units are required to change their activation values smoothly, the components of the target sequence are usually defined as discrete, possibly binary events. The gradual nature of the activations makes it hard for the units to perfectly reproduce such targets. For instance, a given output unit will require some time to reach an activation of 1 coming from an activation of 0, and for all this time the unit will be considered to be wrong – that is, an error signal will be given, that will drive the change in the relevant weights to achieve a stronger activation. But then the unit might be required to get off again, and the previous weight up-

dates would result harmful now. The whole process can result in oscillations in the network behavior. Therefore, care must be taken in the definition of the targets: some tolerance on the activation value that a unit must reach to be considered correct should be allowed. Nevertheless, continuous networks are particularly interesting in modelling cognition as they are more realistic models of the brain.

That's not all, folks: other network classes

The material we have covered in this section does not, by any means, intend to be exhaustive with respect to the ever-growing field of ANNs. New network architectures, and learning algorithm, are being proposed constantly. In this closing part, we briefly mention some other interesting, and historically relevant, ANN paradigms.

- *Associative memories*, or attractor networks, or content-addressable networks. These are actually recurrent networks that differ from the ones we have discussed above because (i) they have symmetric connections, that is for each pair of units p, q $w_{pq} = w_{qp}$, (ii) they are trained in an unsupervised fashion, and (iii) they do not have hidden units. The objective of learning is that of reproducing a particular input pattern when a partial (possibly corrupted) version of it is presented to the network; this can be used, for instance, for image reconstruction tasks. This is accomplished by the network through the storage of the patterns themselves in the form of attractors. An input falling inside a basin is attracted to the corresponding stored pattern, which can be returned as the output of the network. A classical example of an associative memory is the Hopfield network (Hopfield, 1982), which consists of binary units with the sign function as their activation function, that get updated asynchronously (only one unit at each step). The learning rule is basically Hebb's rule (Eq. 4.3), which has the effect of making the weights in the network reflect the correlation between elements in the input pattern. Hopfield also introduced the idea of an energy function $H = -\frac{1}{2}\sum_{pq} w_{pq} y_p y_q$ defined over the network, whose minima are attractors; however, an energy function is guaranteed to exist only if all connections are symmetrical, which is not the case in generic recurrent networks (and, incidentally, in networks of real neurons, which makes this kind of ANNs biologically implausible). *Boltzmann Machines* (Hinton and Sejnowski, 1983, 1986) represent an extension of Hopfield networks where stochastic units are employed (i.e., they can assume either state (1 or -1) according to a probability distribution); also, Boltzmann Machines, differently from other associative memories, do make use of hidden units.

- *Competitive learning networks*. These networks (von der Malsburg, 1973; Grossberg, 1976; Rumelhart and Zipser, 1985) are used for performing unsupervised categorization (that is, clustering) of the input patterns, and are called competitive because output units compete to be the only activated one. In clustering terminology, the winning unit (or, more precisely, its weight vector) is the prototype of the cluster to which the cur-

rent input pattern is assigned. In the most basic setting, we have a two-layer feed-forward network with binary units. The winning unit is the one having larger net input; this is equivalent to say that the unit i is the winner if the (normalized) weight vector $\boldsymbol{w_i}$ is the closest one to the currently presented input vector. Learning in these networks means moving the weight vector of the winning unit even closer to the input: $\Delta w_{ij} = \eta y_i (x_j - w_{ij})$, which again is Hebbian learning (with some weight decay). Many variants and extensions of the competitive learning paradigm have been proposed. For instance, *Adaptive Resonance Theory* – ART (Carpenter and Grossberg, 1987a,b, 1988) – allows for new output units to be created ("enabled") if no existing unit can reasonably account for some new input pattern (in other words, if a novel category needs to be introduced). Feature maps introduce a spatial arrangement of output units so that the notion of similarity between inputs is turned into the notion of spatial closeness of their winning output units. This is achieved by accommodating the output units into a multidimensional array. Feature maps are biologically relevant because they are inspired by topological maps in the brain, like the retinotopic map. A popular example of feature maps are the *self-organizing maps* – SOM (Kohonen, 1982) – which extend competitive learning by updating also the weight vector of neighboring units of the winner, in a way that is proportional to the distance of such neighbors from the winning unit. In this way, most weights are partially dragged toward the winning position, thus preserving distance relations.

- *Autoencoders*. These are special types of feed-forward networks that are trained to reproduce the input they receive. As no additional external information about the output is needed (the goal is to reproduce the input, so the network already possesses all the information it needs), it is possible to regard these networks as being trained in unsupervised mode, albeit using backpropagation. The usual architecture consists of as many output units as inputs, with a smaller number of hidden units. As the network is forced to re-map the input information onto a smaller space and then reconstruct the output from such internal representation, autoencoders are particularly suitable to be used for data compression (e.g. Cottrell et al., 1987, 1989). The autoencoder approach is strictly related to principal component analysis (PCA): for networks with linear units, it was shown (Baldi and Hornik, 1989) that an autoencoder having N hidden units implements the projection of input patterns onto the subspace determined by the first N principle components of the input space.

- *Radial basis function networks*. In these networks (Moody and Darken, 1988, 1989; Poggio and Girosi, 1990; Platt, 1991) hidden units make use of a radial basis function as activation function, most commonly a (normalized) gaussian function. This provides a unit with a specific *receptive field*: this means that the unit will respond maximally to inputs that are close to the center of its receptive field (that is, input vectors that are most similar to the vector representing the mean of the gaussian) and gradually less to those that are further away. Weights from these hidden units to the output layer can be trained in

a supervised way (for instance, by using the delta rule – Eq. 4.9). Unsupervised methods can be employed, on the other hand, to find the appropriate parameters for the gaussians. The idea of a receptive field is very biologically motivated, as, for instance, neurons in the primary visual cortex are known to be tuned for specific directions of lines in a scene.

4.2.2 Artificial Neural Networks in modelling cognition

The use of artificial neural networks as models for mental processes (Thomas and McClelland, 2008) dates back to the 1980s. In particular, numerous modelling efforts were carried out by the PDP (Parallel Distributed Processing) Research Group, whose members are also responsible for the influential text (McClelland et al., 1986) that formalized connectionism. In this section we will present some historical examples of connectionist models of cognitive processes, and discuss the main features of this modelling approach.

Connectionist models are used, as most models in cognitive psychology, to investigate higher-level cognition: they have been applied for modelling various aspects of memory, language, and attention. As connectionist models, as we have seen, are endowed with learning mechanisms for acquiring the task at hand (unlike, for instance, logic-based models) they are also good candidates for studying developmental issues, such as the effects of disrupting learning processes on the performance of the resulting cognitive system.

Connectionist models can be seen as existing at the algorithmic level of the hierarchy defined by Marr (1982): in fact, they necessarily make input and output representations explicit, as well as the algorithm for mapping the former into the latter. In this regard, they have more explicative power than other cognitive modelling paradigms, such as Bayesian modelling, which is situated at the highest level of analysis (the computational one)[10]. Even though the inspiration of ANNs is definitely brain-rooted, the analogy should not be taken too literally. There are profound differences between the (very) simplified artificial neuron and the biological one, as there are between the learning algorithms that have been designed for ANNs and the way learning occurs in the brain (which is still not completely understood). ANNs are necessarily *simplifications* of real neural networks, as simplification is a powerful means for isolating some high-level features that can be more easily understood if lower-level details are left in the shadow. In this sense, the critiques on the biological implausibility of some aspects of connectionism are not to be emphasized, and certainly should not discourage the use of this modelling tool. However, it is important to take these critiques into account, as the strive for more biological plausibility (not to be intended as strict faithfulness, though) goes in the direction of building more and more explicative models. Let us also notice, in passing, that debates on neural plausibility have interested also logic-based accounts, especially on the basis that they usually disregard the largely parallel nature of computation in the brain,

[10]For a head-to-head comparison and commentaries on emergentist approaches – connectionism and dynamical systems – and probabilistic approaches to cognitive modelling, see the dedicated Special Issue of Trends in Cognitive Science, 14(8), 2010.

and typically require chains of rule derivations whose length is hardly compatible with time constraints deriving from human reaction times.

In discussing the biological plausibility of connectionist models, we should start by remarking that the behavior of the artificial neuron itself is just an approximation of the behavior of the real cell. In fact, in the brain neural information is coded through *spike trains*: these are sequences of action potentials (all-or-none events) that are produced with a certain frequency. The actual information is therefore not in the spike itself (all spikes are equivalent), but in the frequency of a spike train. In a connectionist neuron, the output of a neuron is its activation value, which can be binary or real-valued, but either way is a single value. Subsequent activation values are processed separately, not as part of a spike train. Nonetheless, this does not make the connectionist neuron implausible: in fact, the activation value can be seen as an approximation to the real cell's firing rate, so that the highest the activation value, the highest the firing rate, and therefore the highest the response of the neuron to the current input.

The main source of concern for advocates of biological plausibility lies in the learning mechanism. Whereas Hebbian learning and the delta rule are considered to be plausible as they only make use of information that is local to the synapse (whether the pre- and post-synaptic neurons have fired), the backpropagation algorithm is not. This learning rule is not local, in that the necessary information (the discrepancy between actual output and target) may reside far away from the connection being updated: the error signal has to be passed backwards through the connections (and, if batch learning mode is assumed, signals have to be accumulated and stored in the connections themselves). This backward passing of the error signal is considered to be highly implausible from a biological point of view. However, modifications of the original backpropagation algorithm have been proposed that have more solid biological ground: for instance, the *GeneRec* algorithm (O'Reilly, 1996) approximates the error signal by the difference in activation states spread from the output units via feedback connections. Note that, on the other hand, the delta rule has been shown to be equivalent to the Rescorla-Wagner law formalizing classical conditioning (Rescorla and Wagner, 1972) and, as such, is deemed a biologically plausible learning rule. Other problematic aspects of learning algorithms in ANNs include the source of teaching signals (at least for some cognitive task), as it is not clear how information about desired outputs can be given to the brain, and the fact that learning can determine changes in the sign of weights: biologically, this would correspond to an excitatory synapse becoming inhibitory, or vice versa, and it is highly implausible.

Nonetheless, connectionist models have proved their usefulness in advancing our knowledge of the processing properties of the neural system. Some examples will be given shortly, based on their historical relevance, and drawn mainly from the literature on modelling language tasks. Although different models can embody different assumptions about the process they describe, all connectionist models share the same founding principles, namely, that:

- the brain operations are performed in *parallel*, by sets of simple units that simultane-

ously elaborate incoming signals, integrating different sources of information in an interactive way;

- processing in the brain is based on *distributed* representations, that is neural codes are made of the activity of many units;
- learning occurs by *extracting statistical relations* between the observed inputs and outputs characterizing a task.

The major potentiality of connectionist modelling lies in the homogeneity of this approach: although the modelled domains can vary, the same general principles are retained. If such models are consistently able to provide sensible explanations for the observed behavioral phenomena within each particular domain, then these same general principles can be taken to be approximations of the actual principles of brain computation as a whole, rather than specific assumptions about given, specialized sub-modules.

Some examples

The Interactive Activation model

One of the first connectionist models of a cognitive process was the *Interactive Activation model* – IA model (McClelland and Rumelhart, 1981; Rumelhart and McClelland, 1982). In fact, given the above properties, this could be considered a *pre*-connectionist model, in that:

- localist representations are used, in place of distributed ones;
- there is no learning of the network weights: they are hand-coded.

Using localist representations can be more convenient in some cases, and it is especially useful for representing inputs and outputs. However, they are considered to be the equivalent of grandmothers cells (Gross, 2002), and as such their biological plausibility is debated (e.g. Bowers, 2009; Plaut and McClelland, 2010; Bowers, 2010).

However, it is nonetheless an ANN-model, and it has been an influential one, especially in the psycholinguistic field. The IA model was introduced to model letter perception within a word, or a generic letter string. Experimental data show that, when subjects are asked to name a letter inside a string, they are faster if the string is a real word, rather than a random sequence of letters. A similar facilitatory effect has also been reported for legal pseudowords[11]. To provide an explanation of these experimental facts, McClelland and Rumelhart (McClelland and Rumelhart, 1981; Rumelhart and McClelland, 1982) proposed a model where detectors for letter features, for letters, and for whole words are represented by single units (localist coding), organized in a hierarchy of interconnected levels. Connections can be

[11]Non-existent words have been called alternatively pseudowords, or nonwords. Here, we adopt the convention of using *pseudowords* to refer to strings that are pronounceable according to the rules of the considered language (e.g. RINT), and *consonant strings* to denote unpronounceable strings (e.g. RKNT).

either excitatory or inhibitory; bidirectional connections (both excitatory and inhibitory) exist between levels, while only inhibitory connections are allowed within a level. These latter implement the mechanism of lateral inhibition: since only one unit should be active within a group (for instance, units representing all possible letters for a given position), the one having strongest activation gets to inhibit all the others through these connections. The general architecture of the system is sketched in Figure 4.6. The word detectors level is reminiscent of the logogen model proposed by Morton (1969), where a logogen was a device encoding semantic, orthographic, or phonological information about a single word. Each logogen was characterized by features that were "activated" each time a word possessing them was read, leading to a process of evidence accumulation for that word; if the amount of evidence overcome the logogen threshold, the word was recognized. In the IA model, however, processing is not thresholded but occurs in a cascaded fashion: that is, as long as activation is present at one level, independent of its amount, it is passed to all connected layers.

The dynamics of the IA model is rather simple. Each unit p has an activation value $y_p(t)$, computed at discrete time steps, which depends on both the net input from other units, and its resting state r_p. For word units, the resting state depends on word frequency: the highest the frequency of a word, the highest the base level of activation for the corresponding node. Activation for each unit is also subject to decay with rate α_p:

$$y_p(t + \Delta t) = y_p(t) - \alpha_p \left(y_p(t) - r_p\right) + e_p(t) \tag{4.29}$$

where e_p represents the effect of the net input $net_p(t)$ on the node:

$$e_p(t) = \begin{cases} net_p(t)(M - y_p(t)) & \text{if } net_p(t) > 0, \text{ that is the net input is excitatory} \\ net_p(t)(y_p(t) - m) & \text{otherwise, that is the new input is inhibitory} \end{cases} \tag{4.30}$$

Here, M and m represents, respectively, the maximum and minimum activation level for a unit. Notice that, as mentioned, the connection weights in the network are fixed; in particular, all connections between two given layers have the same weight[12].

The basic assumption for this model is that perception is carried out by transforming the input (in this case, an orthographic input) into progressively higher levels of abstraction; moreover, information at the different levels interact (hence the name of the model), both in a bottom-up and in a top-down fashion, to successfully perform the task. Activations from the feature level spread to the letter level, from here to the word level, and then again to the letter level. Within each level, processing occurs in parallel. Finally, the model gives an output (the letter in a specific position) based on the response strength of nodes at the letter level.

The model was successful in simulating a set of behavioral results, including the facilitatory effect in recognition of letters embedded in a word or in a pseudoword, vs. random strings. Whereas the facilitatory effect for words was, to some degree, hand-coded in the

[12]That is, for any two units u_1 and u_2 such that u_1 belongs to layer l_1 and u_2 belongs to layer l_2, $w_{21} = c$, with c constant.

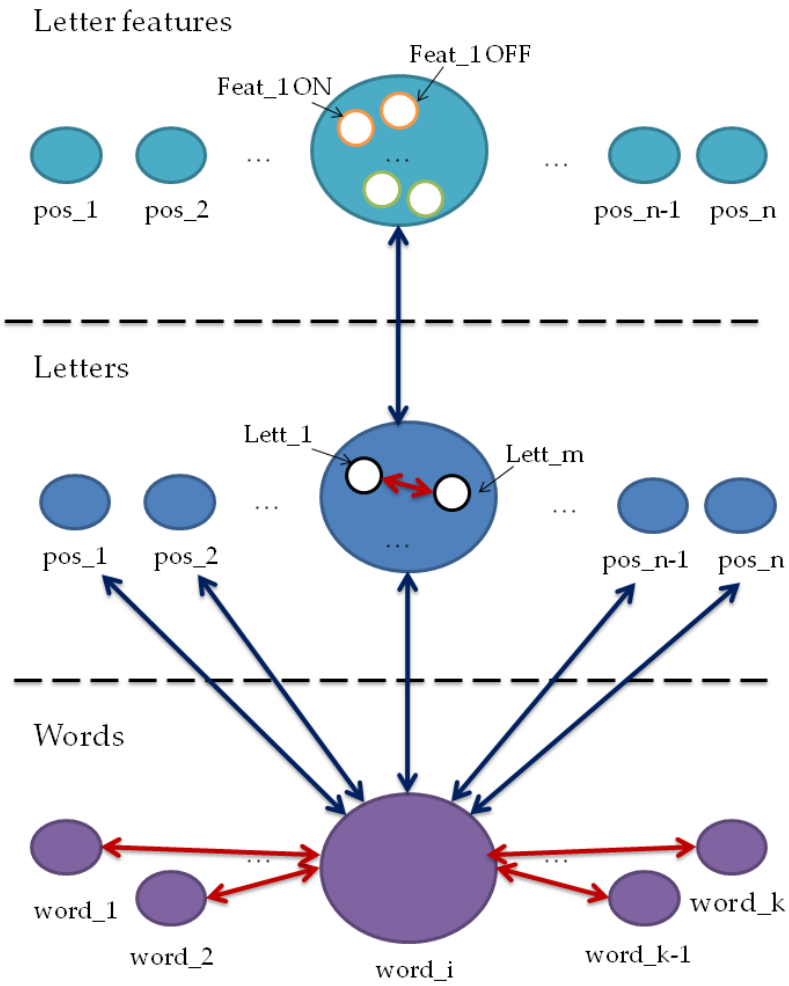


Figure 4.6: A schematic representation of the Interactive Activation model of letter perception. The feature and the letter levels have separate detectors for each word position. Each letter feature (portions of lines composing the character) has two detectors associated: one for denoting its presence, and one for denoting its absence. At the word level, we have one unit for each of the 1,179 4-letter words known by the model. Dark blue arrows represent bidirectional connections, that can be both excitatory and inhibitory. For instance, a word node sends excitatory connections to all its letters in the correct positions, and inhibits all the others. Red arrows are used for inhibitory-only connections. In the simulations described in (McClelland and Rumelhart, 1981), however, word-to-letter and letter-to-letter inhibitory connections were disabled (their weights were set to 0).

model, because words provide excitatory input to their constituent letters, directly increasing their activations, the effect for pseudowords can be regarded as a side-effect of the statistical information inserted in the model. Letters in a pseudoword (e.g. "bame"), in fact, do not receive a strong activation from a single word node; however, they do receive excitatory input from similar (neighboring) words (e.g. "fame", "name", "same", ...), that share one or more letters in the correct positions with the pseudoword. The collective effect of these activations is strong enough to reinforce the target letter node. This cannot happen for a letter within a random string, as its overlap with existing words is much more negligible. In the spirit of good modelling, moreover, the IA model also made a testable prediction: that is, the facilitation effect in the model was observed also for unpronounceable pseudowords that, however, shared some degree of similarity to real words (e.g. "scme", that overlaps in 3 letters out of 4 with "same", "some", "acme"), for the same reason given above. In (Rumelhart and McClelland, 1982) a behavioral experiment for testing this prediction was described, whose resulting data actually agreed with the model performance.

The past tense model

We have mentioned how parallelism, distributed representations, and focus on learning are among the main distinctive features of connectionist models. These are all embodied in another influential model proposed by the PDP group: a model for past tense formation in the English language (Rumelhart and McClelland, 1986). This task is interesting because in English most verbs have a regular past tense – the "-ed" suffix is appended at the end of the verb root –, but some verbs require different constructions (e.g. build $\rightarrow$ built; read $\rightarrow$ read; come $\rightarrow$ came; go $\rightarrow$ went). This feature of English grammar (that has a parallel in the orthographic depth of this language, see Chapter 5) intuitively leads to postulate the existence of two mechanisms for past tense formation: one that takes care of regular verbs, and just applies the "ed" rule; and one that stores present-past pairs for irregular verbs to be retrieved (not computed) when needed.

The past tense model of Rumelhart and McClelland (1986) proposed a different interpretation. The main component of the model is an associator network (two-layer feed-forward network) with 460 input units and as many output units. Output units are binary, and have a probabilistic activation function:

$$P(y_i = 1) = \frac{1}{1 + e^{-(net_i - \vartheta_i)/T}} \tag{4.31}$$

where, as usual, y_i, net_i, ϑ_i are the activation state, net input, and threshold, respectively, of unit i, and T is a parameter called temperature that controls the variability of the units response: for large values of T, $P(y_i = 1) \rightarrow 1/2$, and thus high variability is obtained; conversely, for small values the responses tend to get deterministic and to approximate the behavior of linear threshold units. Incidentally, we notice that the activation function in Eq. 4.31 is the one used in Boltzmann machines as well (see Section 4.2.1).

For learning, the perceptron algorithm (Rosenblatt, 1962, see Section 4.2.1) was used. The training set was composed of (verb root; past tense) pairs. The phoneme string corresponding to the verb root was represented as a distributed pattern of activation over the input units. Each unit represented a given Wickelfeature, that is a triple of phonemic features – organized along four dimensions, e.g. whether the phoneme is an interrupted or continuous consonant, or a vowel – for a phoneme and its two adjacent neighbors in a string. Each phoneme was represented by the activation of 16 Wickelfeatures, and the whole verb root was represented by the simultaneous activation of all the Wickelfeatures in the verb[13]. The same code is used for the output. In order to translate this distributed representation into the phoneme sequence representing the computed past tense of the verb, the model was also endowed with a decoder (but no learning occurs at this stage).

Apart from the goal of achieving good accuracy in the past tense production task, the authors explicitly targeted one developmental finding. It was thought at the time that children go through three phases in past tense acquisition[14]. In the first phase, they make little use, albeit correctly, of verbs in the past tense, and these are mostly high-frequency, and irregular ones. In the second phase, they start using the past tense more frequently, especially for regular verbs; they can also generalize the "-ed" rule to non-existing verbs, and make errors on irregular verbs that were correctly used in the previous phase. In the last phase, finally, they can correctly form the past tense of both regular and irregular verbs, with occasional errors on the latter ones.

In the simulations of the model, the 506 most frequent English verbs were used. The model was first trained on the 10 most frequent verbs, to simulate the first phase of past tense acquisition described above. Then, the mid-frequency verbs were added to the training set. Finally, the remaining low-frequency verbs were used to test the generalization ability of the model. The pattern of experimental findings delineated above was correctly reproduced: in the first phase, the model (like children) achieved good performance on few, high-frequency, mostly irregular verbs; when new verbs were introduced, the performance on irregular words dropped, but got almost flawless – although less precise than performance on regular verbs – by the end of training (U-shaped curve of learning). Generalization to new verbs was good, and the analysis of the errors made by the network also revealed similarities with the kinds of errors made by children (e.g. regularization errors).

The success of the model shows, therefore, that explicit representation of rules (with separate handling of exceptional items) is not necessarily required to explain seemingly rule-based behavior (*if* "verb is regular" *then* "add -ed"). Rather, such behavior can derive from one cognitive mechanism where no explicit rules are encoded, and regularity is only a side-effect of a distributed representation of the statistical structure of the cognitive domain. The ability of

[13]The employed input representation actually involved other Wickelfeatures as well, to create a *blurred* version of the input, in order to promote generalization.

[14]This theory was subsequently challenged by experiments (e.g. Marcus et al., 1992) suggesting that these three phases are not experienced globally, for all verbs, but are rather restricted to some verbs only; this theory is referred to as *micro U-shaped development* (Plunkett and Marchman, 1993).

the network to generalize derives from the fact that input representations are partially overlapping (a verb shares one or more phonemes with other verbs), and therefore knowledge acquired about one verb can also be used for other similar verbs – and this *super-positional* effect is stronger for items that share most statistical structure with the rest of the corpus. It is this statistical structure that is extracted by the network and distributed among its connections.

This example also highlights the value of connectionist models in investigating developmental issues, by analyzing how performance changes in different stages of learning.

Word sequence processing in a simple recurrent network

In the technical overview of ANNs (Section 4.2.1) we introduced the Elman network (Elman, 1990), which falls into the class of simple recurrent networks (SRNs); as we said, SRNs, owing to context units that reproduce the activation of hidden units at the previous processing step, have the ability to retain some memory of the past input and, therefore, to exploit temporal correlations in the data. In the original paper, the new network architecture was actually applied (besides a sequential version of the XOR problem) to language processing tasks – being language one of the human activities most dependent on time. The tasks are presented in order of increasing difficulty.

In the first linguistic task, strings in an artificial language were fed sequentially, letter by letter, to the network. The artificial language consisted of three consonants and three vowels; each letter was represented as a 6-bit vector, where each component coded for a given phonemic feature (e.g. being a consonant/a vowel). The strings were built by randomly generating sequences of consonant and then applying rules that appended specific vowel sequences after a given consonant. The network (having 6 input units, 6 output units, and 20 hidden and context units) was trained to predict, at each time step, the next letter in the sequence. The network successfully learned the task, in that (a) vowels were detected with high accuracy, whereas (b) consonants could not be reliably predicted. However, the network was able to predict that a consonant (not a vowel) was to be presented at the next step: in fact, in trials when a consonant should be predicted, the overall error was large, but the error for the output component coding for the feature consonant/vowel was low.

The second task used in (Elman, 1990) was more complex, as it employed sentences of real English words, rather than random, artificial strings. More specifically, 200 sentences of different lengths were generated using a vocabulary of 15 words, concatenated, and represented as a seamless sequence of 4,963 5-bit vectors, each representing a letter. The network (5 input and output units, 20 hidden and context units) was then trained on this corpus, with the same task as before: predicting the next letter. Although much of the sequence cannot actually be predicted, the pattern of error showed some regularity, in that the maximum error was associated with predicting the first letter of a word, and then error decreased for subsequent positions in the word. This shows that the network was able to extract some degree of statistical regularity in the input; that is, it could learn that some letters have high time correlation.

It also shows that such model is partially able to perform parsing of the letter string, as words happen to be those segments in the input sequence starting with a peak in the error.

In the final task, the SRN was required to predict the next word in a simple sentence, either 2- or 3-words long. This task is expected to be very challenging for the SRN, and for connectionist models in general, as *"[K]nowledge of word order might require symbolic representations which are beyond the capacity of (apparently) nonsymbolic PDP systems"* (Elman, 1990, p.195). There were 29 different words, that could belong to different classes of nouns or verbs; words were combined to respect the correct order in English (i.e. noun-verb-noun, or noun-verb). Each word was represented by a 31-bit binary vector having only one non-zero component (so that word vectors were all orthogonal); the input sequence was a concatenation of 27,354 such vectors. The implemented model had 31 input and output units, and 150 hidden and context units. Since there is no way to reliably predict the occurrence of any word in the input sequence, a different measure of performance than absolute accuracy should be used to evaluate the performance of the network. The best approximation it can be expected from the network is to activate output units to an extent that is proportional to the likelihood of the corresponding words to occur next in the sequence. The comparison of the likelihood values computed on the input sequence with output activations revealed that indeed the network had learned to successfully approximate the prediction task. Additional analyses on the internal representations developed by the network suggested how it could have learned to do so. Elman (1990) computed the mean vector of hidden units activations over all presentations of the same word; therefore, 29 vectors were obtained, that were fed into a hierarchical clustering algorithm. The resulting dendrogram showed that hidden representations for nouns were closer to each other than representations for verbs, for instance. In other words, classes of words (or, more correctly, words in their context, as hidden units combine input and context information) were discovered by the network – although the network did not know the meaning of those classes; it just used them to determine what item was likely to follow at the current point in the input sequence. Class information was extracted based on the time correlation between words in the input sequence, rather than from input representation themselves: in fact, word vectors were orthogonal and did not code for any specific word feature, like grammatical class.

Even though the example networks reported in (Elman, 1990) were not intended to be models of language processing tasks, they nonetheless produced a behavior that, in some cases, was in line with experimental data on the topic, as for instance the relative predictability of word classes with respect to single words. Interestingly, this work pointed out how some degree of word semantics (at least at the level of rough classes based on similarity) may be derived from the mere presentation of words in sequences; in subsequent work (Elman, 1993), this concept was extended to syntax as well. These results increased the popularity of the connectionist approach for modelling language acquisition and processing, as witnessed by several subsequent works on these topics (Christiansen and Chater, 2001).

A recurrent neural network for memorizing sequences

The exemplary models reported thus far were all concerned with language-related tasks. However extremely interesting, language is not, by any means, the only domain to which connectionist modelling has been applied. Another significant line of work has investigated the mechanisms involved in *memory*: for a last example, therefore, we have chosen to illustrate one model in this domain. We picked the model for serial memory by Botvinick and Plaut (2006) for two reasons: (a) because it explicitly targets sequential processing, which is crucial in the model we propose in Chapter 6, and (b) because it employs a recurrent neural network. This last point, besides being once again relevant for the material presented in Chapter 6, allows us to give a natural conclusion to the ideal walk we took in this section through model architectures of increasing complexity: from a fixed-weight, localist network of spreading activations (the IA model) to a two-layer feed-forward network (the past tense model), to an Elman network, and finally to a more general recurrent network.

The model presented in (Botvinick and Plaut, 2006) consists of n ($n \in [13, 43]$ in different simulations) input units, 200 hidden units organized in one layer, and n output units. Besides the usual feed-forward connections, the hidden layer is connected (in all-to-all fashion) to itself, and gets feedback from the output layer. As a result, at each step the internal representation over the hidden layer depends on the current input, and on both the previous internal representation and output.

Hidden units employ a logistic activation function, whereas output units use the softmax function, that is:

$$y_i = f(net_i) = \frac{e^{net_i}}{\sum_{k=1}^{n} e^{net_k}} \tag{4.32}$$

with k ranging over the output units. Using this activation function has the effect to induce competition among output units: as the overall output layer activation is bounded to be 1, each unit must compete with the others to be the most active (and therefore winning) one.

The addressed task is known as immediate serial recall (ISR): given a sequence of items presented in subsequent time steps, the subject is asked to name them in the right order right after the completion of the sequence presentation. This task was implemented by turning on one input unit at each time step, with each unit representing one item, and then activating a special input (recall cue) to signal the network that the sequence has ended. Output representation is specular to input. The network was trained on sequences of different lengths (from 1 up to 9) using BPTT; an error function different from the ones introduced thus far was employed:

$$E = \sum_{i=1}^{n} y_i \log \frac{d_i}{y_i} \tag{4.33}$$

This is called *divergence error*. Interestingly, if a softmax activation function is used in com-

bination with this error measure (as in this case), it has been shown (Rumelhart et al., 1996) that the resulting trained network approximates maximum a posteriori classification. At each time step after the recall cue has been turned on, the response of the network was identified to be the item whose corresponding unit had maximum activation (excluding the special unit used for signalling the end of recall).

The model was evaluated against a set of experimental findings, showing the plausibility of this recurrent NN-based approach to account for the mechanisms involved in serial memory. What is most interesting to discuss here is how the network managed to achieve the ISR task. This question translates to asking how the activation patterns over the hidden units (i.e., the internal representations) were organized. Intuitively, there are two dimensions that need to be coded: which elements appear in the input sequence, and in which order. The analysis of the hidden layer activations revealed that such information was coded in a super-positional way: that is, the representation of the sequence was a combination of representations for the single component items. Moreover, the representation for a single item simultaneously coded for both the item identity, and its position in the sequence, and representations for similar items in similar positions tended to be close. Botvinick and Plaut (2006) also offered a geometric interpretation of the ability of the network to store the entire sequence and, at the same time, produce only one item per step ("output gating" facility). We have already seen (Section 4.2.1; in particular, see Fig. 4.2(a)) that computing the net input to a unit via an inner product corresponds to projecting the incoming activation vector onto the weight vector for that unit. The magnitude of such projection (i.e., the size of the net input) is proportional to the degree of similarity of the two vectors; vectors that are more aligned will produce a stronger activation of the output unit. This follows trivially from the definition of inner product:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \|\boldsymbol{a}\| \, \|\boldsymbol{b}\| \cos \gamma \tag{4.34}$$

where γ is the angle between the two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. As $\cos \gamma$ is maximum when $\gamma = 0$, that is, when the weight vector and the activation vector are perfectly overlapping, in this case the output unit will be strongly driven to large activation; when the two vectors are orthogonal, on the other hand, the activation vector will have no effect (it will be "invisible") on the output unit. The authors, then, analyzed the degree of similarity (as measured by $\cos \gamma$) between the representation vector for each element in the input sequence (the activation vector) and the weight vector for the correct output unit at each time step: they found that the network performed a sort of rotation of the activation vectors, so that at each step only the representation of the item in the correct position was well "visible" (large $\cos \gamma$) to the output layer. This mechanism allowed the network to efficiently store all information about the input sequence while passing to the output layer, at each step, only the relevant element.

This work also allows us to make a general point about the interpretation of the concept of memory in connectionist accounts. According to a traditional view, memory can be interpreted as a warehouse where items (single memories) are placed. When a specific memory needs to be retrieved, the warehouse is accessed, the memory located, and possibly moved

to another place where it can be further processed. In connectionist models, memory is the model itself; more precisely, the network embodies task-specific memory, that it has acquired while learning to perform that task. In the above model for short-term serial order memory, the memory trace for a sequence provided by the environment is encoded in the form of network activations. These, in turns, are produced by specific weights that have being learned to the purpose of achieving an efficient "storage". So, acquired knowledge (experience on the task) is instantiated in the connection weights; and it can interact with current, contextual knowledge, which combines with connection weights to produce unit activations. In this way, knowledge and memory retrieval are the results of computation, rather than storage access.

4.3 Conclusion

In this chapter we have provided an overview of computational modelling of cognition, with a particular focus on connectionist modelling. The models we have presented here have allowed us to illustrate the main characteristics of the connectionist approach to modelling (such as the use of distributed representations, the commitment to parallel computation, the ability to learn from statistical relationships in the problem domain, the capability of exhibiting rule-following behavior without needing any explicit rule specification, and to also accommodate exceptional associations), and to highlight some of their potentialities. We have therefore seen how a particular modelling framework can be applied to different tasks and domains. In the next chapter, we will do the reverse, and examine how different modelling approaches, based on different theoretical assumptions, have been applied to the study of a specific sub-domain of cognitive psychology: single word reading. In Chapter 6, finally, our own connectionist modelling work on reading is presented.

Chapter 5

Computational models of single word reading

"Any man who reads too much and uses his own brain too little falls into lazy habits of thinking."
— Albert Einstein, 1879–1955

5.1 Single word reading: data and theories

Single word reading (also referred to as naming) involves mapping an orthographic form (the written word) into its correct phonological form (its pronunciation). While this is mostly trivial in Italian, for instance, it is not in English. In fact, English is a *deep orthography*: this means that the correspondence between orthographic units (graphemes) and phonological units (phonemes) is, in general, not unique. On the other hand, shallow orthographies like Italian display a high degree of regularity in grapheme-to-phoneme correspondences. Therefore, in English we can distinguish between regular words (for instance, MINT, pronounced /mint/), which employ "regular" (in general, the most frequent) grapheme-to-phoneme correspondences, and exception words (for instance, PINT, pronounced /pInt/). Before going any further, it is important to remark that reading is a cognitive process that involves more than just orthographic and phonological manipulations, as reading a word also conveys a meaning. However, the role of semantics in models of single word reading has classically been somewhat ancillary, the focus being more on the mechanisms that support a correct pronunciation of the supplied orthographic items. The role of semantics in naming, moreover, is supposedly negligible at least when facing pseudowords (see Note 11) that can, nonetheless, be pronounced with no particular difficulty. However, even when no explicitly targeted, or fully modelled, a contribution of semantics to reading is usually assumed, both as a side-effect (meaning is implicitly activated when a word is read) and as a source of information that supports the general process of reading. Several other tasks are closely related to reading: for instance, spelling (the mapping from phonology – or semantics – to orthography) and production (the mapping from semantics to phonology). Another related task is lexical decision: deciding whether a given orthographic stimulus is, or is not, a real word.

To study how the human brain performs the mapping between a written form into the corresponding spoken one, behavioral experiments on skilled readers have been conducted, as well as observations on brain-damaged patients[1]. Historically, it has been especially this latter line of studies that offered the most informative, and challenging, data on single word reading. Dyslexia (or alexia) is defined to be a disorder of skilled word reading, and it can be either developmental or acquired (following brain damage). Different forms of acquired dyslexia have been observed in patients, each characterized by a specific pattern of deficits. For our discussion, the most relevant syndromes are phonological and surface dyslexia. A patient affected by *phonological dyslexia* (Beauvois and Derouesné, 1979; Coltheart, 1996) typically shows relatively worse performance when reading pseudowords, as opposed to real words; conversely, *surface dyslexia* (Marshall and Newcombe, 1973; Coltheart et al., 1983; Patterson et al., 1985) is characterized by relatively impaired performance on exception words with respect to pseudowords and regular words. Extreme cases of these syndromes have been described in the literature and have assumed a paradigmatic role in illustrating such double dissociations: patient W.B. (Funnell, 1983), whose reading was mostly spared for real words, and totally impaired on pseudowords; and patients M.P. (Bub et al., 1985) and K.T. (McCarthy and Warrington, 1986), who both suffered from surface dyslexia. While K.T. was a more extreme case as he preserved an almost intact ability of reading regular words and pseudowords, but was severely impaired in exception word reading, M.P.'s deficit on exception words was less dramatic.

Additional characterization of reading has been provided by behavioral experiments, that uncovered a pattern of effects reliably shown by subjects during reading tasks. These include the frequency effect (Forster and Chambers, 1973; Frederiksen and Kroll, 1976), by which frequent words are read faster than infrequent words, and the lexicality effect (Forster and Chambers, 1973; McCann and Besner, 1987), by which real words have lower naming latencies than pseudowords. Regularity in the print-to-sound correspondences also plays an important role: a frequency-by-regularity interaction has been observed (Seidenberg et al., 1984; Taraban and McClelland, 1987; Paap and Noel, 1991), that is, regular words are read faster than exception words, but only for low-frequency items. A related effect (to the degree that consistency can be seen as a generalization of regularity, see farther in this chapter) is the consistency effect (Glushko, 1979; Jared et al., 1990; Jared, 1997, 2002; Treiman et al., 2003): lower naming latencies are associated with words and pseudowords having a consistent body (where by body we refer to the final part of a word, from the last vowel). A consistent body is one that is pronounced the same way in all the words that contain it. It is important to notice here that the concept of consistency can have different interpretations. Glushko (1979) defined it to be a binary variable (if a word has a body that is pronounced differently in at least one other word, then it is inconsistent); according to others (e.g. Jared, 2002), consis-

[1]More recently, the body of knowledge about the cognitive process of reading, initially alimented by data coming from behavioral and neuropsychological studies, has seen significant contributions coming from neuroimaging experiments, whose main aim is to locate the neural circuits involved in specific reading sub-processes. We have offered an account of these data in Chapter 3.

tency is a graded variable that depends on the ratio between friends (words with the same orthographic body that is also pronounced the same) and enemies of a word (same body, but different pronunciation). Finally, serial effects (that is, effects that have been interpreted as consequences of sequential processing) have also been established: namely, the length effect (Frederiksen and Kroll, 1976; Weekes, 1997), whereby long words correspond to longer naming latencies, although for monosyllabic stimuli, the effect is significant for pseudowords only; and the position of irregularity effect (Coltheart and Rastle, 1994; Cortese, 1998; Rastle and Coltheart, 1999), whereby longer latencies are recorded for exception words having the irregular correspondence in the first positions with respect to irregularities later in the word.

The pattern of the collected behavioral effects and the characteristics of the observed reading deficits have led to postulate a variety of cognitive models of reading, with special emphasis on *dual route models* (e.g. Baron and Strawson, 1976; Besner and Smith, 1992; Coltheart, 1978, 1985; Forster, 1976; Morton and Patterson, 1980; Marshall and Newcombe, 1973; Paap and Noel, 1991; Patterson and Morton, 1985b). Mostly based on the double dissociation observed in phonological and surface dyslexic patients, the dual-route theory postulates that reading is supported by two different cognitive mechanisms, responsible for the correct pronunciation of different categories of stimuli: a mechanism for applying conversion rules, therefore apt to pronounce regular words and pseudowords, and a dictionary-like mechanism that looks for the given orthographic item inside a mental lexicon, and returns the corresponding phonological entry (especially important for pronouncing exception words). These two mechanisms are also refereed to as implementing assembled phonology and addressed phonology, respectively. According to this account, phonological dyslexia is the consequence of a lesion to the rule-based system, whereas surface dyslexia follows from damage to the orthographic lexicon. In dual-route accounts, semantics is seen as contributing to reading by providing an indirect way to retrieve phonology from print (reading by meaning), but its role is considered to be secondary with respect to the purely lexical route and the sub-lexical one. In sum, the lexical side of the model can be conceived as a collection of orthographic, phonological, and semantic entries, that are separately stored for each known word (in this it is related to the previously mentioned logogen model). The sub-lexical route computes pronunciations by applying rules that reflect the regularities in the print-to-sound mappings of the language. Coltheart (1978) refers to these rules as grapheme-to-phoneme conversion (GPC) rules. According to this account, for example, the frequency-by-regularity interaction described above can be accommodated by postulating different processing speeds for the two routes, with the lexical one being faster, especially for frequent words: in this way, when a high-frequency exception word is read, the (correct) pronunciation returned by the lexical system reaches the output stage earlier than the regularized response computed by the sub-lexical route (and therefore the lexical response gets to be directly outputted – "horse race" theory (Paap and Noel, 1991) – or has a stronger impact in driving the final output toward itself). On the other hand, if the exception word has low frequency, the speeds of the two routes are similar, and the conflicting outputs determine a higher latency in the response time.

Even though the dual-route theory has been the leading theory of reading ever since it was first formulated, alternative theories have also been proposed, and especially it has been argued that assuming the existence of two separate circuits for achieving skilled reading is unnecessary: a single mechanism would be sufficient to successfully handle all kinds of orthographic stimuli. For instance, lexical analogy models (Glushko, 1979; Kay and Marcel, 1981) propose that the pronunciation of a word or pseudoword is computed based on the pronunciations of neighboring words (that is, words that carry similarity with the target words in any of their segments). This interpretation was strongly influenced by the consistency effect reported by Glushko (1979), especially by the finding that pseudowords are not always pronounced in a regular way, but can make use of the pronunciation of the body of similar exception words. The adjudication between theories is complicated by the fact that behavioral effects that have been pointed out as arguments to support a dual-route theory can be also reinterpreted under a single process point of view. According to a PDP framework, for instance, the frequency-by-regularity interaction observed in human readers can be explained as the result of training conditions. A regular word is, by definition, composed of print-to-sound correspondences that are most frequent in the considered orthography. Since these correspondences are shared by many words in the training corpus, the network can easily learn them, whereas exception words require more extensive training to achieve the same degree of robustness. The regularity effect comes from here. On the other hand, the more frequent a (regular or exception) word is, the more it gets to shape the weights of the network (i.e., the better it is learned), which means that a high-frequency exception word can partially compensate for the penalty associated with its irregularity due to its high frequency inside the training corpus. Basically, the frequency-by-regularity interaction arises from the combination of two levels of frequencies within the word: frequency of words, and frequency of print-to-sound correspondences.

Thus, both dual-route and single-route accounts offer sensible, yet different interpretations of the processes involved in single word reading. Nor the large amount of neuroimaging studies carried out in this field in the last 20 years has provided consistent evidence for or against any of these competing theories (but see Chapter 3).

5.2 Single word reading: computational models

Computational models embodying these contrasting theories have been proposed starting from the late 1980s (Zorzi, 2005; Seidenberg, in press; Plaut, 2005). They greatly contributed to the debate about theories of reading by providing actual implementations of their founding principles. A typical computational model of single word reading aims at reproducing the main behavioral findings reported in the literature, by:

- producing a pattern of computation cycles, and number of errors, for each category of input stimuli, that is comparable to what has been described in the literature for normal subjects;

- simulating the effects of a brain lesion (e.g., by unit removal), and showing that the resulting pattern of results is consistent with what is observed in dyslexic patients.

The results of model simulations are then submitted to statistical analysis to assess their fit with the relevant human data. This can be done by computing latencies (or error rates) produced by the model for different classes of stimuli (e.g. regular vs. exception words) and using a statistical test (typically, analysis of variance, ANOVA) to determine whether the size of the effect is significant; effects that are found to be significant (respectively, non-significant) for human readers should be significant (non-significant) for the model as well. Another approach (regression analysis) consists in computing the correlation between human data and model data on a single-item basis to determine how much of the variance in the human data is accounted for by the model.

In what follows, we will illustrate the main computational models for naming, focusing on those committing to either of the above described theoretical accounts. Three classes of models will be presented: the connectionist single-mechanism models (also known as Triangle models), the dual-route cascaded model (DRC), and the connectionist dual-process models (CDP and its successors). All of them are restricted to the English language, and to the processing of monosyllabic words only (but see Ans et al., 1998). Before looking into the details of these models, however, a mention to an influential precursor[2] of such models, NETtalk (Sejonwski and Rosenberg, 1986), is in order.

5.2.1 NETtalk

NETtalk (Sejonwski and Rosenberg, 1986; Sejnowski and Rosenberg, 1987) is a connectionist network designed to pronounce English words that are provided sequentially, one letter at a time. The output of the network is a sequence of phonemes to be possibly elaborated by a speech synthesizer to actually pronounce the resulting utterance. Although NETtalk was proposed more as an engineering application than as a psychologically-oriented model, it is nonetheless interesting to discuss here, as a pioneering work that showed the potentialities of the PDP approach in modelling tasks in the reading domain.

The architecture is that of a feed-forward two-layer network, with 203 input units organized in 7 groups, a variable number of hidden units (most simulations used 80 units), and 26 output units. Units in the input layer represent single letters in each of the 7 possible positions of the input string. The network uses a tapped delay line scheme, whereby letters composing the input word are shifted of one position to the right at each subsequent processing step. The network has to produce the phoneme corresponding to the letter in the central (i.e., fourth) position; however, neighboring letters are nonetheless processed by the network, and constitute the context for the current center letter, therefore supporting its correct pronunciation. The output units represent articulatory features, so that each phoneme is simultaneously coded by the activation of a set of output units (that is, phonemes have a distributed representation).

[2]We have already described (Section 4.2.2) another founding model in this domain: the Interactive Activation model by McClelland and Rumelhart (1981).

The network was trained (by backpropagation) on both natural running text, and a corpus of isolated words. Letters and phonemes were manually aligned to provide the training pairs; for letters constituting multi-letter graphemes (like SH), the corresponding phoneme was associated with the first letter, and a special symbol (−) representing silence was given as target for the second letter. The trained network was able to achieve very high levels of performance (95% and 98% on the two datasets, respectively), and generalization was good as well (around 77% for both datasets, but improving with more extensive training). Sejnowski and Rosenberg (1987) experimented with different variants of the network, revealing that (a) the larger the input window the better the performance – enlarging the context for a letter help decide the correct pronunciation; (b) more hidden units yielded a better learning curve, and with no hidden units the performance on the dictionary dataset was no higher than 82% – hidden units allow the network to discover nonlinear relations among letters; (c) the network was quite robust to mild damage (that is, random perturbation of the trained weights) – this is also a characteristic of the human brain. Finally, the authors also analysed the internal representations for single letter-to-sound correspondences by submitting to a hierarchical clustering algorithm the set of vectors representing the mean activitation value of each hidden unit associated with each correspondence. The clustering procedure showed a marked separation between representations for vowels and for consonants, thus providing some insight into the mapping strategy learned by the model.

NETtalk represented a very first step toward the implementation of models of word reading. Differently from subsequent connectionist models, that adopted mostly parallel representations for both input and output, NETtalk was strictly sequential. This line of work was pursued also by Bullinaria (1997) and Plaut (1999). In (Bullinaria, 1997) an evolution of NETtalk was presented which was able to perform alignment of inputs and targets (e.g. for the word BOOK, possible alignments are, for instance, B ⇒ /b/, OO ⇒ /u/, K ⇒ /k/, and B ⇒ /b/, O ⇒ /u/, O ⇒ /k/,K ⇒ -) by itself, choosing the set of targets that minimized the training error. The network was trained on three tasks: reading, spelling, and past tense production, achieving perfect performance on the training data and good generalization, although the fitting to experimental effects was not completely satisfying (for instance, no frequency effect was found).

Plaut (1999) proposed a sequential model for reading that included the possibility for the network to refixate a problematic portion of the input string, that is, to focus attention on a letter/grapheme for which pronunciation was not correct. This is a recurrent network having 27 input units for each of 10 positions inside a word; the target of fixation at each step is the letter at the third position. The output layer consists of 36 phoneme units, to be activated in the correct sequence. In between, there is a hidden layer of 100 units with autorecurrent connections, and with connections to and from a layer of position units, indicating where fixation is currently at; the hidden layer also receives connections from the output layer. Whenever the network fails to pronounce the correct phoneme, or to produce the correct next fixation position, a refixation occurs. This resulted in a large number of refixations early in

training, that decreased over time as the network learned to master the training set. The number of refixations performed by the network was taken as an analogue of human reaction time, and using this measure a reliable pseudoword length effect was identified. Traditional effects of frequency and regularity were also found. Finally, corrupting the activation at the level of input units produced a pattern of results consistent with the syndrome of letter-by-letter reading (Dejerine, 1892). In brief, the model retains the sequential nature of phoneme production as in NETtalk, whereas proposing a more parallel approach to input processing: in fact, in those trials in which only one fixation occurs, the whole word is processed in parallel; when two or more fixations are required, the input is treated in a similar way as in NETtalk (i.e., it is shifted).

5.2.2 The Triangle models

The first real attempt at providing a computational model of the cognitive process of reading was the PDP model by Seidenberg and McClelland (1989). The SM (from the initials of the authors) model bears notable similarities with the past tense model (Rumelhart and McClelland, 1986) (see Section 4.2.2). Both models adopt, in fact, the same representational scheme based on Wickelfeatures[3]; and, more importantly, both are set to learn mappings in a domain that is defined *quasi-regular*, in that, besides a majority of regular correspondences, exceptional ones exist as well. Whereas dual-route models explain this feature of the English orthography by postulating the existence of two different mechanisms for handling rule-compliant items and exceptions, the SM model (and PDP models in general) is based on the idea that the parallel interaction between different sources of information (orthographic, phonological, semantic) in a neural network is sufficient to account for skilled reading of any kind of orthographic stimulus.

The framework proposed in (Seidenberg and McClelland, 1989) includes orthographic, semantic, and phonological units, with each group connected with the others via a hidden layer, in the typical triangular structure that gave the name to the family of models fathered by the SM model. It is therefore postulated that reading can be accomplished either by a direct orthography-to-phonology pathway, or by a semantically mediated one. The theoretical formulation also includes the contribution of a layer coding for word context, connected to the semantic layer. The architecture of the full proposal is sketched in Figure 5.1(a). However, only a subset of this framework was actually implemented. The implemented SM model (Figure 5.1(b)) is a two-layer network with an input layer encoding the orthographic representation of the stimulus (400 units), a hidden layer of 200 units, and an output layer (460 units) coding the phonology of the input stimulus; the units employ the logistic activation function. Input units also receive feedback connections from the hidden layer so that the original orthographic input can be recreated. This is equivalent to having a strictly feed-forward network with two sets of output units, corresponding to the phonological output and to the

[3]The scheme is identical for phonological representations. As for orthographic representations, each unit codes for the presence in the input string of one of the 1,000 letter triples to which that unit responds.

orthographic output, respectively (Figure 5.1(c)). The network was trained by standard back-propagation (with a momentum term) to both produce the pronunciation of the input word, and to reproduce the input pattern. This latter output of the network was used to simulate the task of lexical decision; however, for reasons of space, we will concentrate our discussion on simulations of reading only[4].

The network was trained for 250 epochs on 2,897 monosyllabic words derived from the Kučera and Francis (1967)'s corpus and from published experimental studies. The probability of a word W being presented to the network for training during an epoch was proportional to the logarithm of its frequency:

$$p(W) = K \log(\text{freq} + 2) \tag{5.1}$$

The use of log-frequencies, which are much more compressed in range than the original values, was preferred both to reduce the number of training epochs required to expose the network to the whole corpus for a sufficient number of times, and to get closer to the reduced range of frequencies experimented by beginning readers (that is, children).

In order to determine the fitness of the model in accounting for the relevant behavior effects, an internal measure that could be considered an analogue of naming latencies in human readers had to be identified; however, since all outputs are given in a single step of processing, in the SM model there is no such direct analogue. Seidenberg and McClelland (1989) chose to use as a measure of latency the phonological error score, that is the value of the error function used by the training algorithm over the phonological units. The rationale behind this choice is the assumption that a low error score is associated with a clearer response, which, under time pressure, would drive the articulatory mechanism faster than a more blurred phonological code. Phonological error scores were also used to determine the model accuracy, by comparing the actual pattern of activation obtained on the phonological units to a set of plausible phonological outputs (including the target, and other relevant classes of possible erroneous outputs – as, for instance, the regularized pronunciation for an exception word). If for a given orthographic item the best fit among these comparisons was obtained in correspondence to the actual target, then the network was said to have correctly pronounced that item. Using this measure, the SM model was found to correctly read the 97,8% of the words in the training corpus, both regular and exception ones. It also showed the classical frequency-by-regularity interaction and a good fit with experimental data taken from a range of behavioral studies. An important contribution of the SM model lied in that it showed how a wide set of effects could be explained in terms of graded spelling-to-sound consistency: rather than assuming that regular words (together with pseudowords) and exception words belong to two non-overlapping classes, and as such they need to be processed by two different, dedicated mechanisms, the connectionist paradigm invites to interpret orthographic stimuli as varying along a continuum of consistency, which can operate at different orthographic levels (like

[4]This applies to the other models we introduce in this chapter as well: the DRC model (Coltheart et al., 2001) can also simulate the task of lexical decision, but we will restrict our discussion to the naming simulations.

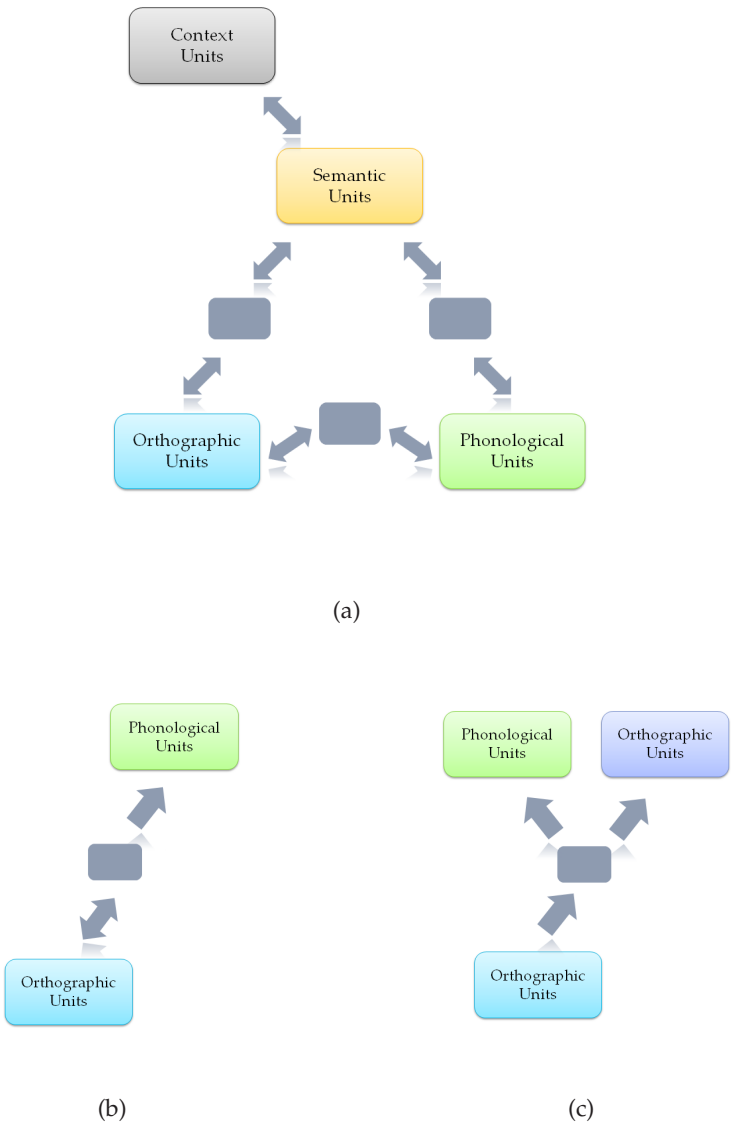


Figure 5.1: The Seidenberg and McClelland (1989) model of reading aloud. Boxes represents sets of units; gray boxes are used for hidden units. Panel (a) shows the theoretical framework for the model: orthographic, phonological, and semantic distributed codes interact to support skilled word and pseudoword reading; additional information can be provided by the context in which the word appears. In panel (b) the actually implemented model is shown. Panel (c) illustrates the equivalent interpretation of the implemented model as a strictly feed-forward network, where we consider the orthographic output to be computed by a duplicated set of orthographic units, distinct from the ones coding for the input.

word bodies and spelling-sound correspondences). Since consistency directly affects connection weights by reason of the learning mechanism, the more consistent an item is, the stronger its internal representation is, leading in turns to lower error scores (and naming latencies).

However, the SM model is limited in many respects (see for instance Coltheart et al., 1993). The employed input/output representations are complex and do not allow for a straightforward understanding of what the output of the network, in terms of word pronunciation, is. The model was considered to have correctly pronounced an item if the phonological error score for the correct pronunciation was the lowest among a set of alternative responses – but not all possible responses were compared, so the chance existed that some incorrect responses actually had a lower phonological error score than the correct one. However, the most important limitation of the model resides in its poor performance in pseudoword reading (Besner et al., 1990): this suggests that the network was not able to develop a sufficient degree of generalization.

This seminal work had various evolutions. For instance, Bullinaria (1996) investigated how an implemented semantic system could be integrated in a connectionist network, so as to pursue the direction drawn by the original SM formulation. The proposed architecture consists of an orthographic, phonological, and semantic layer, each connected to each other via a set of 500 hidden units (a layer for each mono-directional connection). Units in the orthographic layer represent graphemes, phonological units represent phonemes, whereas the representation on the semantic units is arbitrary: each of the 513 words in the training corpus was coded by a random binary distributed code. Each word was presented to the network for 150 time slices: at each presentation, the word representation according to one variable (e.g. semantics) was fed as an input, while the other two representations (the orthographic and phonological ones) were used as targets. Latencies were extracted by setting a threshold on the activations of the phonological layer. Performance on the words in the training set and on the tested pseudowords was virtually perfect, and promising preliminary results in the simulation of behavioral effects were reported. However, the model was still limited, both because of the reduced size of the training corpus, and because semantic representations were completely unconstrained, therefore limiting the range of experimental findings that could be simulated.

The real successor of the SM model was the PMSP model (Plaut et al., 1996). While the network architecture stayed basically the same with respect to the SM model, the PMSP model significantly improved its performance by adopting a different representational scheme for both input and output. In fact, the authors argued that, since poor performance on pseudoword reading is a distinctive sign of a lack of generalization power in the network, input/output representations should be used that enhance the discovery of regularities across the dataset. The representations used in the SM model, on the contrary, tended to disperse knowledge, because the same letter-sound correspondence, e.g. L $\Rightarrow$ /l/, had to be learned (i.e. coded in weights) for each different context (triple) in which it occurred. Thus, knowledge was not being extended to different contexts.

The new representational scheme involves phoneme and grapheme units, both organized in three sets of units corresponding to the onset (initial set of consonants), vowel, and coda (final set of consonants) of a word. All possible graphemes in a word (both single letters and multi-letter graphemes) are input to the network; and the output layer activation represents all the phonemes in the word. Phonotactic constraints, implemented as mutually exclusive groups of phonemes, allows for the easy readout of the pronunciation computed by the output units, since phonemes in a cluster can usually be assembled only in a given order; an additional phoneme unit is introduced to distinguish between the cases when different orderings are possible (e.g. /ps/ and /sp/).

Simulations were conducted on a feed-forward network having 105 input units, 100 hidden units, and 61 output units. The network was trained by backpropagation to learn a corpus of 2,998 monosyllabic words; logarithmic word frequency, differently from the SM model, was not used to sample the training set, but was instead multiplied to weight updates, so that more frequent words had a larger impact on weight changes. After 300 training runs, performance on word reading was at ceiling, and pseudoword reading accuracy results were in line with experimental findings; thus, the change in representational scheme was actually successful at improving generalization in the network. Effects of frequency, consistency, and their interaction were also replicated, although the model also exhibited a consistency effect among high-frequency words that is not found in human readers (however, this effect disappeared when the network was trained by using actual, rather than logarithmically transformed, frequencies). Plaut et al. (1996) also offered an analytic account of how frequency and consistency effects arise in a connectionist network: both high-frequency of presentation, and high degree of consistency in spelling-to-sound mappings have the effect of diminishing the error for a word, since they strengthen the relevant, correct connections, thus increasing the activation of correct output units, and reducing the activation of the others. However, sigmoidal units saturate for large enough inputs, so if one of these factors is sufficiently high, the other one will have negligible impact. For this reason, consistency effects are not usually observed for high-frequency words.

Furthermore, an additional simulation on a recurrent network was reported in (Plaut et al., 1996). This is a network having continuous dynamics (discretized for computational purposes), with layers sized as in the feed-forward network but, in addition, with feedback all-to-all connections from the output layer to the hidden layer, and recurrent connections among hidden units. This network was trained by the continuous version of BPTT over 2 time intervals, with 5 ticks per interval (the number of ticks was then increased in the final stages of training for getting a finer approximation to continuous dynamics); for testing purposes, the network was considered to provide its response when the output units had settled to a stable state (that is, their average change in activation was no larger than a fixed threshold). In this way, each input item was associated with the number of ticks required for computing a stable response to it: this provided a more natural analogue of human naming latencies. Performances on both words and pseudowords, and fit of the simulation data to standard

behavioral effects, were again good. Interestingly, the network developed attractors with a compositional nature. In fact, especially for regular words, the network learned that there exist partially independent correspondences between orthographic and phonological clusters, such that, for instance, only (or mainly) the graphemes in the word coda determine the correct pronunciation of the phonemes in the coda itself (while they have negligible impact on the phonemes in the onset). As a consequence, the basins of attraction for regular words were organized along three dimensions (corresponding to onset, vowel, and coda) so that the attractor for a word lied at the intersection among the three sub-basins that composed it. Such compositional structure supported pseudoword reading as well – pseudowords can be think of falling into intersections, not corresponding to any real word, of sub-basins for words –, whereas exception words had less compositional attractors.

Taken together, the simulations run on the PMSP model were able to successfully cover a wide range of behavioral effects, thus supporting the connectionist view of reading as an essentially homogeneous process that takes advantage of consistencies across orthographic, phonological, and semantic sources of information for handling all types of strings. However, the fitting of the model data to the human ones turned out to be not accurate as far as serial effects, like the length (see Spieler and Balota, 1997) and position of irregularity effects (see Coltheart et al., 2001) were concerned. This was taken by some as a proof that a parallel system like a connectionist network cannot account for serial effects. Later work (Plaut, 1999) suggested that explicit modelling of sequentiality in input acquisition (visual processes) and output production (articulatory processes) could help a parallel network account for these effects (see above). The proposed approach was also interesting in that, in principle, it could be used to process also polysyllabic words. However, the model was still rather limited in the range of effects it could simulate, leaving space for developments along this direction (see Chapter 6).

Besides addressing mature, skilled reading, connectionist models have also been employed to simulate data from dyslexic patients. Three forms of dyslexia have been targeted: surface dyslexia, phonological dyslexia, and deep dyslexia (Coltheart et al., 1980). Deep dyslexia is characterized by severely impaired pseudoword reading (for this reason, the hypothesis was advanced that deep dyslexia might be a more extreme version of phonological dyslexia (Glosser and Friedman, 1990; Friedman, 1996)) and by semantic errors – i.e. reading a different, but semantically-related, word than the one that is presented – visual errors, and visual-then-semantic errors. Deep dyslexia has been simulated by lesioning a trained network in (Hinton and Shallice, 1991) and (Plaut and Shallice, 1993). Hinton and Shallice (1991) designed an attractor network to map orthographic representations into semantics; attractors corresponded to word meanings, with similar meanings having attractors close in space. Damage to the network determined occasional shifts of the network activity toward an incorrect attractor, but located near the correct one: this gave rise to the typical semantic errors observed in deep dyslexia. Visual errors were observed, too. Plaut and Shallice (1993) expanded the model by adding computation of the phonology of words from semantics, and

provided evidence for the double dissociation, observed in the literature, in reading concrete and abstract words: while a lesion to the "direct" pathway (from orthography to semantics via a set of intermediate units) produced an impairment in abstract word reading, a severe lesion to the "clean-up" pathway (from semantics to itself via a set of clean-up units) produced the reversed pattern of deficit (whereas mild damage to this portion of the network caused no relative deficit). This finding shows how double dissociation observed behaviorally can be explained by different patterns of lesion within the same processing system, without having to assume that different specialized modules (in this case, for concrete and abstract words) exist.

Surface dyslexia was simulated in (Patterson et al., 1989; Patterson, 1990) by lesioning the SM model: different proportions of units and connections were removed, and the resulting patterns of errors were, to some degree but not completely, consistent with data from surface dyslexic patients. More convincing results were obtained in the fourth simulation presented in (Plaut et al., 1996). To this end, the authors simulated the contribution of semantics to naming by providing an additional input to phonological units to drive them toward the corresponding target. In interpreting how semantics contribute to reading, the hypothesis put forward is that division of labor occurs between the purely phonological pathway and the semantically-mediated one. As semantics helps in reading words, during the course of learning the direct pathway does not need to master all by itself the whole training corpus, but will come to partially rely on the additional information provided by semantics. As a result, the phonological pathway will tend to concentrate on the easiest input/output relations (because they are characterized by high frequency or consistency), while more difficult words will depend more on the contribution of the semantic pathway. Under this hypothesis, surface dyslexia would result from a lesion to the semantic pathway, which has the effect of isolating the phonological pathway; since this is not fully competent on its own, errors on difficult words (i.e. low-frequency exception words) are expected. To test this hypothesis, the feed-forward version of the PMSP model was trained with an additional input to the phonological layer simulating semantic contribution, having value

$$S = g\frac{\log(\text{freq}+2)t}{\log(\text{freq}+2)t+k}, \tag{5.2}$$

where t is the training epoch, and g and k are constants governing the asymptotic growth of the signal. The semantic signal was therefore proportional to the frequency of the word, and increased with training time. The input value S to a phonological unit was positive if the target was 1, negative otherwise. After training the network under these conditions, a full semantic lesion was simulated by removing the external signal. The pattern of performance on the different classes of items (regular and exception words, of high and low frequency, and pseudowords) was consistent with typical data from dyslexic patients. It was also advanced that different patterns of performance observed in patients might be explained by a different premorbid division of labor between the two pathways: some readers may tend to rely

more than others on semantic mediation, and therefore would present a more severe deficit following damage of the semantic system; additional simulations on this point were reported in (Plaut, 1997).

As for phonological dyslexia, the connectionist approach would suggest that it may be the result of damage to the direct phonological pathway, whereas the semantic pathway remains relatively spared[5]. However, since no full implementation of the semantic pathway was available for both the SM and PMSP models, a direct simulation for phonological dyslexia was not possible. Subsequent work addressed specifically the modelling of semantics in reading (Harm and Seidenberg, 2001, 2004). The proposed model is a continuous recurrent network (see Fig. 5.2) where orthographic units map to both phonological and semantic units, both through direct connections and through hidden layers. Phonology and semantics are also connected to each other bi-directionally via two sets of hidden units; finally, two sets of hidden units receive and send connections to the phonological and semantic layer, respectively. Orthographic input is coded by letter units organized in position-dependent slots; phonological and semantic representations are in the form of binary vectors of features (e.g., for semantics, the *living being* feature). This attractor network was at first trained to learn the mapping between phonology and semantics in both directions (to account for the fact that learning to read occurs after phonological and semantics representations are already in place), and then training on orthography was added. In (Harm and Seidenberg, 2001) the effect of graphemic complexity in phonological dyslexia was studied by lesioning the phonological system with multiplicative noise on the connection weights. The investigations reported in (Harm and Seidenberg, 2004) were more focused on effects arising when computing the meaning of words, rather than their pronunciation. However, the performance of the model on the naming task was consistent with the classical effects of frequency and consistency, and pseudoword reading was in line with human data. Therefore, this model showed the viability of introducing semantics into connectionist models of naming.

5.2.3 The Dual-Route Cascaded model of reading aloud

As mentioned above, the dual route account is based on the assumption that two different mechanisms exist that support skilled reading: a rule-based system, responsible for the correct pronunciation of regular words and pseudowords; and a lexical system, that stores for each known word its phonological form, so that exception words can be correctly read by accessing the corresponding entries in such lexicon. Dual-route theories have been most influential in the field of psychology of reading, especially as they provide a straightforward explanation of dissociations in dyslexic syndromes in terms of selective damage to either of the two routes, and a natural, common-sense way of accounting for differences in exception and regular word reading, and pseudoword reading, as relying on mechanisms having dif-

[5]However, it has also been advanced (Plaut et al., 1996) that phonology itself, rather than the pathway from orthography to phonology, may be impaired in this syndrome, since phonological dyslexia has often been found to be associated with less reading-specific phonological deficits.

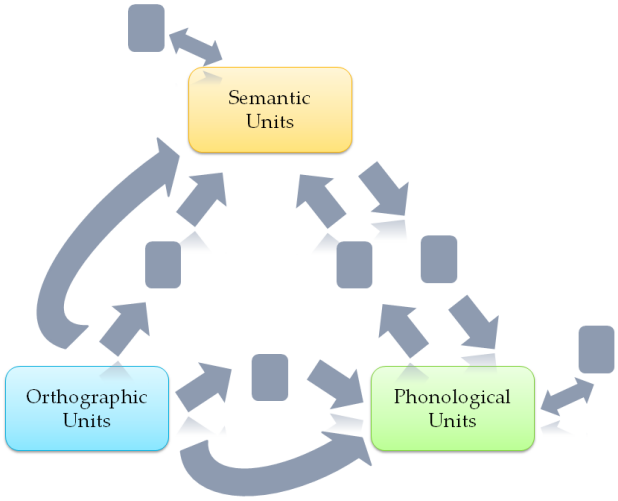


Figure 5.2: The Harm and Seidenberg model. Each gray box represents a set of hidden units.

ferent properties (such that, for instance, pseudoword reading is slower than word reading because the serial rule-based system is slower than the lexical one).

The most popular implementation of a dual route theory is the dual-route cascaded (DRC) model by Coltheart et al. (1993, 2001). Preliminary explorations on the GPC side of the model were reported in (Coltheart et al., 1993), where an algorithm is described that extract the most frequent grapheme-to-phonemes rules, based on the same corpus of words used in (Seidenberg and McClelland, 1989), and the main features of the future model were exposed. A full implementation was then given in (Coltheart et al., 2001) (see Fig. 5.3), although it must be noticed that the semantic system, while part of the theoretical framework, was not implemented (like in the SM and PMSP models);); also, the GPC rules were no longer learned, but hand-coded. The DRC model combines three main sub-components: the lexical route, the sub-lexical (or GPC) route, and the phoneme output system.

The lexical route is based on the IA model by McClelland and Rumelhart (1981) (see Section 4.2.2); as we have seen, activation in the IA model is passed through levels in a cascaded, rather than thresholded, way, and this motivated the DRC name. The original IA model was extended to be able to process words of variable length, up to 8 letters. As in the original formulation, there are feature detectors, connected to letter detectors (organized in positional groups of 27 units), which in turns are connected to word detectors. Connections can be excitatory or inhibitory (see Fig. 5.3, and the above description of the IA model). For each of the 7,891 words in the model corpus, an orthographic entry and its corresponding phonological entry[6] are present in the orthographic and phonological lexicon, respectively, and corresponding pairs are linked by positive connections. The phonological lexicon is connected to phoneme units, organized in a positional manner likewise the letter units (44 units for each

[6]Notice that the phonological entries total 7,131 units, since the corpus contains also a number of homophones.

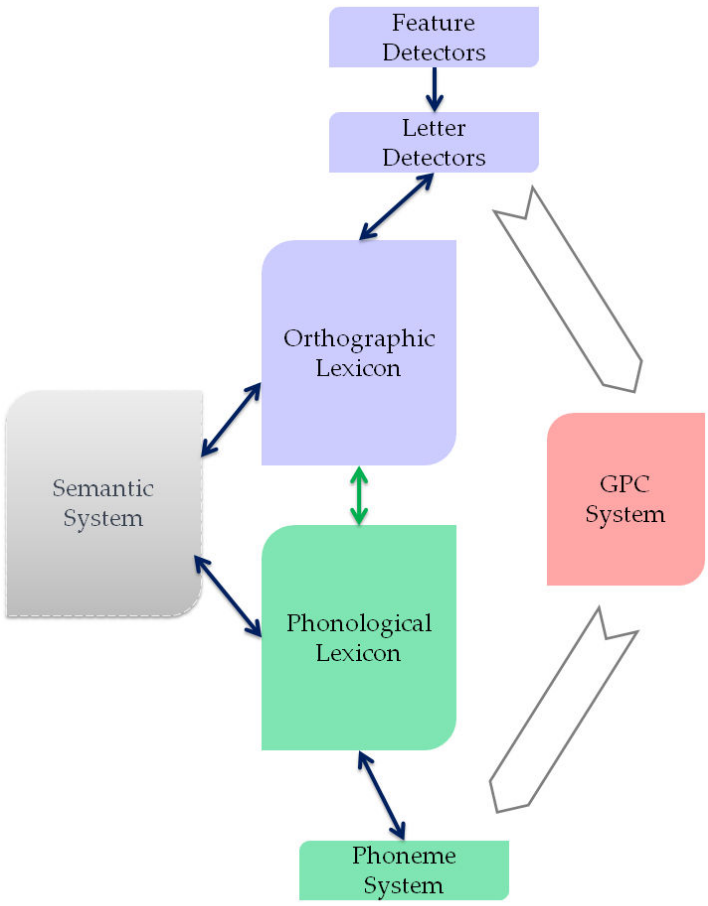


Figure 5.3: The Dual-Route Cascaded (DRC) model. The semantic system is shown as part of the theoretical framework, but was not actually implemented. The blue portion of the model is an adaptation of the IA model by McClelland and Rumelhart (1981); combined with the green components, it constitutes the lexical (non-semantic) route of the model. Blue arrows are used when both excitatory and inhibitory connections exist between two levels (see the discussion about the IA model for further details). The sub-lexical route is implemented as sequential application of conversion rules; different forms for arrows were used for the sub-lexical route to emphasize that these do not represent connections among units, but only denote the flow of computation.

position). The activation dynamics of the units in the lexical route is given by:

$$y_p(t + \Delta t) = (1 - \alpha_p)y_p(t) + (e_p(t) \cdot act) \tag{5.3}$$

where *act* (activation rate) governs the speed of activation change; all activation values are kept within 0 and 1. All other parameters in this equation have the same interpretation as in Eq. 4.29. Notice however that the resting state r_p does not appear in this equation; recall that in the original IA model the resting level of activation for a word unit was proportional to the frequency of that word. In the DRC model, the contribution of frequency is incorporated in

net input computation:

$$net_p(t) = \sum_q w_{pq} y_q(t) + F_p, \tag{5.4}$$

where

$$F_p = \left(\frac{\log_{10} freq_p}{\log_{10} \max_k freq_k} - 1 \right) \cdot fs \tag{5.5}$$

Here, $freq_p$ represents the actual frequency of the p-th word and fs is a constant used to weight the importance of the frequency factor. The lexical part of the model is therefore a localist, connectionist model, where activation spreads from the feature level down to the phoneme system. The presence of feedback connections allows for information to flow upwards, too. As in the original IA model, there is no learning in this model; all knowledge is hand-coded.

The GPC route operates by applying serially, letter by letter from left to right, the appropriate rules of conversion representing the regular grapheme-to-phoneme correspondences in the English language (e.g. L $\Rightarrow$ /l/), as coded by the modellers. This route receives new input (i.e. a new letter) from the feature and then letter levels that it shares with the lexical route at constant delays; precisely, it receives the first letter at processing cycle 10, and each subsequent letter every 17 cycles. The GPC route is therefore more similar in nature to classical rule-based symbolic models.

In the phoneme system the results of the computation carried out by the two routes are combined. The application of a GPC rule to each letter determines the production of a sequence of phonemes; and the activation in the lexical route of the phonological entry for the input string spreads to its component phonemes. The net input to the corresponding units in the phoneme system is therefore composed of a lexical contribution, and a sub-lexical one. Such contribution is proportional to the overall strength of the GPC route (fixed parameter) and the average activation of the letters to which the employed rule was applied. At each processing cycle, the maximally activated phoneme in each position is retrieved: if, for each position, this has an activation value over the response threshold (0.43), and the last activated phoneme us the word terminator, then this final pronunciation is returned, and the number of cycles needed to get to this response is taken as naming latency for that word.

It can be easily seen that the model is rich in free parameters (31 in total). These were chosen empirically, by looking for the combination of parameters that would account for as much data as possible. A potential risk of this approach is that it might overfit data: in other words, it might account for the data because it was *made* to account for those data, but fail when faced with new data.

When tested on the words in its corpus, the DRC model produced 27 errors on exception words while on speeded naming mode, and just 1 error when reading at leisure (response threshold at 0.70). On pseudoword reading, it produced 75 errors on a corpus of 7,000 items.

The model successfully reproduced the standard effects of frequency – as a consequence of Eq. 5.4 –, lexicality – because the sub-lexical route is forcedly slower than the lexical one –, and frequency-by-regularity interaction (for the reasons exposed above, see p.129). As a direct consequence of its architectural assumptions, the DRC model also succeeded in simulating the position of irregularity effect: in fact, since phonemes from the GPC route are generated in a sequential way, the first one (the leftmost phoneme in the word) will reach the output system early on, and therefore perform a more sustained inhibition activity than later phonemes on the (correct) lexically produced phoneme; the effect gets smaller as we move right within the word. Another serial effect, the observed length-by-lexicality interaction, was also reproduced; this is because pseudowords can be correctly read only by the GPC route, which takes a fixed number of cycles to process each letter in a string, whereas words can be read by the parallel lexical route that is not sensitive to length.

Neighborhood effects in naming were considered as well. For a given string, the variable N is commonly used to identify the number of words that are orthographically similar to that string: more precisely, it is the number of words that can be obtained from the given string by changing one letter. It has been found that a larger N has a facilitatory effect in reading both words and pseudowords (for a review, see Andrews, 1997). In the DRC model, the N effect was significant only for pseudowords, and was determined by the additional activation to the phoneme system provided by neighbors inside the orthographic lexicon; for words, such influence was contrasted due to the high value of the parameter governing lateral inhibition between word units, so that when a word unit is fully activated it strongly inhibits all other units. For similar reasons, the experimentally observed facilitation for pseudohomophones, that is, pseudowords that are pronounced like a real word, e.g. BRANE (McCann and Besner, 1987), especially for those orthographically similar to their base words, were replicated by the model. As for consistency effects, the model succeeded on replicating the effect on words, but not on pseudowords.

Finally, simulations of surface and phonological dyslexia were carried out. Surface dyslexia was simulated (in unspeeded naming mode) by increasing the fs parameter in Eq. 5.5, which in turns made the constants F_p (Eq. 5.4) more negative (as $F_p \in [-fs, 0]$); this choice was meant to simulate a lesser degree of excitability of entries in the orthographic lexicon, which makes lexical access more difficult. As for phonological dyslexia, the pseudohomophone advantage observed for some patients was simulated by increasing the number of cycles needed for a new letter to be submitted to the GCP route (from 17 to 27).

The modelling approach taken by the DRC model is most distant from the connectionist principle of a model *"in which as little as possible of the solution of the problem is built in and as much as possible is left to the mechanisms of learning"* (Seidenberg and McClelland, 1989, p. 525). Rather, the DRC model pursued a data-fitting approach: that is, the model was built so that it could account for the largest pattern of the considered experimental data. It cannot but be acknowledged that the DRC model was able to simulate an impressive range of experimental findings. However, it might be argued that the model provides little insight into the actual

mechanisms implementing human reading. How are rules acquired, for instance? And how could neurons implement the application of if-then-else rules? As knowledge in the DRC model is encoded completely by the modeller, these questions are hardly addressed.

5.2.4 The Connectionist Dual-Process models

Thus far, we have seen connectionist implementations based on a single-mechanism assumption, and a less computationally homogeneous approach that embodies the dual-route theory of reading. However, a third way is possible: that is, a fully connectionist implementation of a dual-route theory. This was realized in the Connectionist Dual-Process (CDP) model (Zorzi et al., 1998), and its successor, CDP+ (Perry et al., 2007)[7].

The CDP model proposed in (Zorzi et al., 1998) is composed of a sub-lexical part named the two-layer assembly (TLA) network, a lexical system, and a phonological decision system (PDS) where responses from the two systems converge and the final pronunciation is decided (Fig. 5.4(a)).

The TLA is a two-layer feed-forward network. Input units are organized in 8 slots of 26 units each, representing a letter in a given position inside the input string; the first three slots are for the onset letters, and the remaining ones for the word body. Words shorter than 8 letters are assigned to the appropriate slots in a leftmost fashion, and for empty positions no unit is activated. Output units have similar organization, with 44 units in each of 7 slots (3 for the onset and 4 for the body). The activation function used by the output units is a sigmoidal function:

$$f(x) = \frac{1}{1 + e^{-(x-1)T}}, \tag{5.6}$$

where the parameter T (set equal to 3) is called temperature, and is used to control the slope of the sigmoid, and $f(0) \approx 0$ as a result of the additional term in the exponential function. Since the TLA has no hidden units, the delta rule could be used for training; a corpus of 2,774 monosyllables was used, all of them presented an equal number of times to the network (that is, there was no frequency-based sampling, nor weight updates depending on word frequency). The limited power of a two-layer network does not allow for the whole corpus to be perfectly mastered; as a consequence, the TLA alone, while it was able to read regular words and pseudowords almost perfectly, could read only few exception words. It was therefore presented as the connectionist implementation of the GPC route postulated by dual-route theorists.

As for the lexical part, two alternative implementations were suggested. In the first one, the TLA is simply augmented with a set of hidden units between the input and the output layer, so that two sets of connections exist: the direct ones, from orthographic units to phono-

[7]Recently, an extension of the model to disyllabic reading, CDP++, has been implemented (Perry et al., 2010). This was accomplished by duplicating both the orthographic and phonological buffer so that two syllables could be accommodated, and introducing two sets of nodes that determine stress assignment to either syllable, both in lexical and sub-lexical mode. Although this is an interesting extension, we are focusing here on monosyllabic word reading, and for this reason we will not discuss this model in further details.

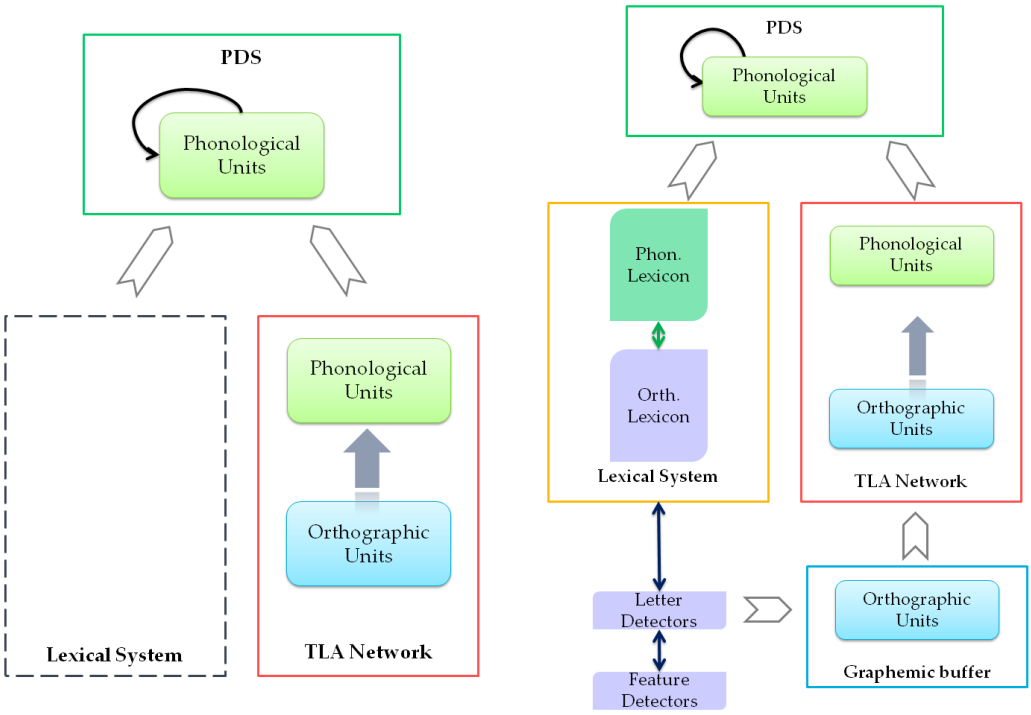


Figure 5.4: Panel (a) shows the architecture of the first version of the CDP model. Note that the lexical part of the model is not implemented. In panel (b) the CPD+ model is shown, where a graphemic buffer has been added to the sub-lexical route, and the lexical route of the DRC model has been incorporated.

logical units, and the mediated ones, going through the hidden layer. The hypothesis is that the direct connections should specialize for regular spelling-to-sound mappings, with the indirect connections providing the additional resources for learning a distributed lexicon (and therefore being able to name exception words); evidence that this happened, at least to some degree, are reported (but note that the number of hidden units had to be limited to prevent the mediated route to be fully competent on its own). Such a network thus performed a form of division of labor, similar in principle to what was described in (Plaut et al., 1996) relatively to the direct and semantically-mediated pathways. The second implementation for the lexical route, which was preferred to the former since it allows the output of the two routes to compete in a dynamic way in the PDS, consists in simulating the contribution of a lexical route, simply as a signal proportional to word frequency that excites correct phonemes and (more strongly) inhibits all the others.

The outputs of the two routes are combined in the PDS. This has the same number of units as the phonological layer of the TLA, but in addition it hosts inhibitory lateral connections between units in the same slot, and excitatory auto-feedback connections for each unit. The input to a unit in the PDS is given by summing the outputs of both routes, that are made

to gradually increase in time to reach their actual value through a multiplicative ramping factor, and the internal input to the unit (excitatory contribution of the previous activation value for that unit, and inhibition from all the other units in the slot). This determines a competitive dynamics between candidate phonemes for the same position: eventually, at most one phoneme for each position will reach the response threshold and enter the final pronunciation. As the competition in the PDS takes time to resolve, the number of processing cycles before a response to an input string can be provided is taken to be the latency of the model.

Reading performance on the training corpus and on pseudowords was consistent with that of skilled readers. Classical behavioral effects were replicated: frequency effect (directly modelled in the simulated lexical route), its interaction with regularity (a stronger input from the lexical route helps resolving conflicts at the PDS level that arise from the co-production of alternative phonemes characteristic of exception words), lexicality effect (input to the PDS is provided by one route only and therefore phoneme units take more time to reach the response threshold). Consistency effects were produced by the TLA network[8]: the reason for this is that the TLA does not directly output the regular phonemes, but other alternative phonemes are produced as well, albeit with a lower activation value; longer latencies for inconsistent strings are a result of the ensuing competition dynamics in the PDS. Finally, the pattern of surface dyslexia of patient K.T. was simulated by increasing the ramp factor so that the lexical output built up at a slower rate.

The main problem with the CDP model was that it was unable to simulate serial effects; it also lacked a full implementation of the lexical route. The subsequent version of the model, CDP+ (Perry et al., 2007), addressed these shortcomings by augmenting the TLA network of the CDP model with a graphemic buffer (Houghton and Zorzi, 2003), that submits serially parsed graphemes to the network, and combining the resulting sub-lexical route with the lexical route implemented in the DRC model (Fig. 5.4(b)). Input representation in the sub-lexical system is changed from single letters to graphemes, to improve generalization and, therefore, pseudoword reading. A graphemic parser process looks for the most active letters in each position, and at regular intervals moves from left to right an attentional window that spans over 3 letters; the letters inside the window are assembled into graphemes, with precedence given for multi-letter graphemes. Graphemes are therefore produced serially and inserted into the graphemic buffer in the appropriate positions. Input to the TLA is therefore serialized. Training occurred for the TLA only, and consisted of a first phase when single grapheme-phoneme pairs were used as a training set, followed by training on the full training corpus (7,383 orthographic forms); differently from the CDP model, learning was made frequency-sensitive by multiplying weight updates by a function of word frequency. A pronunciation was given

[8]Interestingly, also in the DRC model the observed consistency effect on words was in fact produced by the sub-lexical route, as shown by the authors by running the model when this route had been turned off. However, what the GPC route produced was not a consistency effect, but rather the so called *whammy* effect (Rastle and Coltheart, 1998), whereby five-letter strings yield longer naming latencies if their pronunciation consists of 3 phonemes vs. 5 phonemes. In the GPC route of the DRC model this happens because, as a consequence of the sequential application of the conversion rules, earlier produced phonemes that turn out to be incorrect compete with the new, correct ones, thus retarding the model response.

when a settling criterion over the phonological units was met. The new model was able to improve its predecessor by simulating serial effects, and by explaining a larger amount of variance in the reaction times of human readers.

The CDP and CDP+ models share similarities with the other two classes of model we have introduced in this chapter. On the one hand, they are connectionist models – at least as far as the TLA and PDS are concerned – where the mapping between orthographic representations and phonological representations is encoded in connection weights. However, it should be noticed that learning only occurs in the TLA; parameters for the lexical system and the PDS are hand-coded. On the other hand, the CDP models are undoubtedly dual-route models, in that they clearly distinguish between a lexical and a sub-lexical route. However, they part from the DRC model in characterizing the sub-lexical process. This is, in fact, held to be fast and parallel, whereas the GPC route is serial and slower than the lexical one; and whereas the GPC system is based on the application of explicit production rules, the TLA implements implicit, learned mapping rules. Thus, in the CDP models a dual-route theory of reading is defended, but with different assumptions than in the DRC model, and it is shown how self-modularization of the two processes could be achieved in a neural network setting.

Although very successful in accounting for human data, the CDP+ model is still limited in that no learning is implemented outside the TLA network, and thus the behavior of the system is largely determined by the hand-coded parameters that it mainly inherited from the DRC model. This is not just a stylistic difference: a model that *learns* to display a similar reading behavior to the one observed in humans only by being exposed to the task, provides a deeper insight into the mechanisms supporting reading that can be interpreted as the result of task requirements and (limited) available computational resources. It also potentially suggests explanations that can be applied to other cognitive tasks as well, as presumably learning procedures in the brain are largely shared among cognitive modules. In this sense, models that learn have a more explicative power.

Let us observe explicitly, in closing, that a fine but important difference between the dual-route and the single-route interpretations of reading lies in the concepts of regularity and consistency. Dual-route theories emphasize the former: the sub-lexical procedure relies on *rules* (either productive, or mapping rules) that capture the regularities in the letter(or grapheme)-to-sound correspondences; irregular items cannot take advantage from such rules. According to the single-mechanism view, on the other hand, reading mainly relies on consistencies in the language at different orthographic levels. In fact, the dichotomy between regularity and irregularity that characterized dual-route theories, is here turned into a consistency continuum, where no sharp distinctions exist among classes of words. Thus, an open question remains as whether our brain segregates what it learns in the processing of different neural circuits, according to this dichotomy, or rather it deals with both highly consistent and exceptional mappings within the same circuit.

5.3 Conclusion

Under many respects, the above computational models have been greatly successful, in that they offered alternative explanations on how reading might be carried out by our brain both in normal conditions and in presence of a dyslexic syndrome, although the debate is still open on which of them can be considered as the most plausible one. While the DRC model has succeeded in simulating a wide range of behavioral effects, it appears to be, to some degree, artificial in its design, and characterized by a too large number of free parameters; moreover, at least in the lexical component, it lacks any form of learning. Similar considerations can be applied to the CDP models. On the other hand, while the parsimony and simplicity of Triangle models are appealing, there are still some effects (the so called serial effects, mainly) that can be hardly reproduced by these models. To date, it is still unclear if these limitations are due to specific architectural choices, and therefore can be overcome with more careful implementations, or are direct consequences of the underlying assumptions.

In (Coltheart et al., 2001) it is argued that good modelling science should be cumulative, that is, each new model should account for at least as much data as its predecessors, and constitute an evolution of previous, unrefuted models such that, for instance, parts of previous models are integrated into the new one. To use the words of Jacobs and Grainger (1994, p. 1329): *"a new model should be related to (or include), at least, its own, direct precursors and be tested against the old data sets that motivated the construction of the old model before testing it against new ones"*.

However, this is not necessarily the only, or the most successful, research strategy. An existing model might well account for 90%, say, of all known results on reading, and yet being unable to explain that last 10% independently of how many extensions of it might be attempted. This might happen, for instance, if the former model was built in a data-fitting fashion, with the explicit aim of accommodating that 90% of data, but, as a consequence, leaving no degrees of freedom for other classes of data outside the original data set. A model, in short, might be "rather good", and yet it might be quite far from the correct model that would account for all data. We can think of it like an optimization problem, where candidate solutions are organized in a tree structure where solutions down the same branch are increasingly better than those at the upper levels: we might then start exploring one branch, and at the end of it find a solution that we claim to be optimal (in our example, a model that accounts for 90% of the data). This solution cannot be improved by going further down this same branch, because we already reached its end. However, it turns out that exploring another branch actually leads to a better solution (maybe the global optimal one), even though we first need to pass through less good solutions that exist upper in the branch (in our example, these would be models that explain 50% of data, for instance)[9].

This view illustrates how this "backtracking" strategy might be useful in modelling: rather than fully committing to an existent model and endlessly refining it, one can try and

[9]Basically, this is the classical conflict between exploration and exploitation.

devise new, different models that might account for less data than precursor models, but open a new direction that can potentially lead to further improvements.

It is the strategy we take here: although the model we will present in the next chapter is a connectionist model that follows the path drawn by Triangle models, it also experiments with new features, such as a threshold mechanism that lets the network determine when it is safe to start a pronunciation, and serial output. In general, we have taken the view that it should be general principles to guide the design of a model, rather than the data we wish to fit: for this reason, the connectionist approach is the more natural to adopt here, as we believe that the principles of distributed representations, parallel computation, and automatic learning of statistical input/output relations offer a set of fundamental assumptions that might well explain the bases of the actual brain computations.

As we are only at the very beginning of this particular branch, we cannot expect this new model to account for a very large set of data, nor we can anticipate whether this branch will turn out to be fruitful or not. Yet we believe that the principles on which our model is based are interesting in their own, and should be further explored to determine how consistent they are with the actual mechanisms implemented by the human brain.

Chapter 6

Toward a new model for single word reading

"The beginning is the most important part of the work."
— Plato, 427 BC–347 BC

6.1 Scope of the work

We have seen in the previous chapter how the "perfect" computational model of single word reading has not been implemented yet (assuming such model will ever be built), as all extant proposals are limited in some respects. In particular, when considering connectionist single-route models the most important weakness lies in their trouble in accounting for serial effects.

Is this a necessary consequence of employing a parallel architecture? If this was the case, then purely parallel models should be dismissed as inaccurate explanations of the reading process, since serial effects are consistently shown by human readers in the literature. However, it might be the case that this inability to replicate serial effects is not inherent to parallel computation, and could be overcome by new careful implementations. It might be advanced that serial effects are induced by peripheral processes concerning visual analysis, attention, and articulation, rather than sequentiality in the application of spelling-to-sound rules like it is postulated in dual-route models. In such case, it should be possible to build a model that incorporates sequentiality at any of its ends (at the input or at the output), but computing the orthography-to-phonology transformation in parallel, as information flows in, rather than in a strictly sequential manner, on a single correspondence basis.

In this chapter we will describe our work aimed at designing one such model: that is, a model that, albeit retaining the parallel nature of the computation, seeks to introduce sequentiality into the picture. In what follows, we will describe the evolution of the model, and the lessons learned during its development. As this is essentially preliminary, exploratory work, the focus has been set on detailing alternative implementation choices and training approaches with the two-fold aim of illustrating the practical difficulties in a modelling work, and giving a sense of the many strategies that can be devised when developing, and trying to improve, a model.

6.2 The working framework

In developing this model, we first established a general framework that constituted our main assumption: namely, that different sources of information (orthographic, phonological, and semantic) combine to support the acquisition of internal, distributed codes for words. Such codes would be *multiple-purpose*: that is, the presentation of, say, the semantic representation of a word to the model would determine the activation of the internal code for that word, which in turns would produce the appropriate phonological and orthographic representation for the word. Depending on the current task, an overt answer from any of these representations can then be returned. In connectionist terms, it is natural to interpret such internal code as an attractor of a continuous recurrent neural network that builds up and gets clearer as input information is processed. A schematic representation of this general framework is given in Figure 6.1. Let us explicitly notice that by assuming such architecture, a truly *single-route* approach is obtained, as one path only can be taken when mapping orthography to phonology (differently, e.g., from the PMSP model, which, while not being dual-route in the classical meaning of the term, still assumes the cooperation of a direct pathway and an indirect, semantically mediated pathway). The idea of an all-purpose code is not new; we will review previous related proposals in the next section.

Sequentiality may be introduced at the orthographic and phonological levels of representation. Phonemes in an utterance are obviously produced in a sequential way, and therefore sequential output on this side of the model is highly desirable. As for orthographic input, sequentiality can be introduced as well, especially for long strings. For shorter strings, an attentive process that gradually modulates the extent and the amount of information that is fed into the system at subsequent time steps could be assumed. The multiple-purpose code for the string being processed would rise in time as information flows in: when the internal code is sufficiently unambiguous and clear, a response can be initiated.

Let us consider, before going any further, what kind of predictions about the behavior of the model we can make regarding serial effects. First of all, a code will get clearer as more information flows in. The model may perhaps attempt a response before all information about the input string (in the reading case, all letters) has been encoded, on the basis of the partial and weak internal code it has computed up to that point – but that could be wrong. Waiting for all information to be consistently encoded into a clear representation would require longer latencies for longer strings. However, the length effect would be less relevant for words, as presumably the network has developed attractors for them, and should therefore be able to produce a correct response even in presence of an incomplete, or blurred, internal code. Therefore, a length effect should be shown that is more significant for pseudowords, consistently with what has been reported in the literature. As for the position of irregularity effect, we may advance that an earlier irregularity would be more harmful in the quick rising of an unambiguous code because it would require additional evidence to be collected (from later portions of the word) to disambiguate it; for later irregularities, conversely, a large portion of the input has already started to be encoded and therefore the resolution of the current

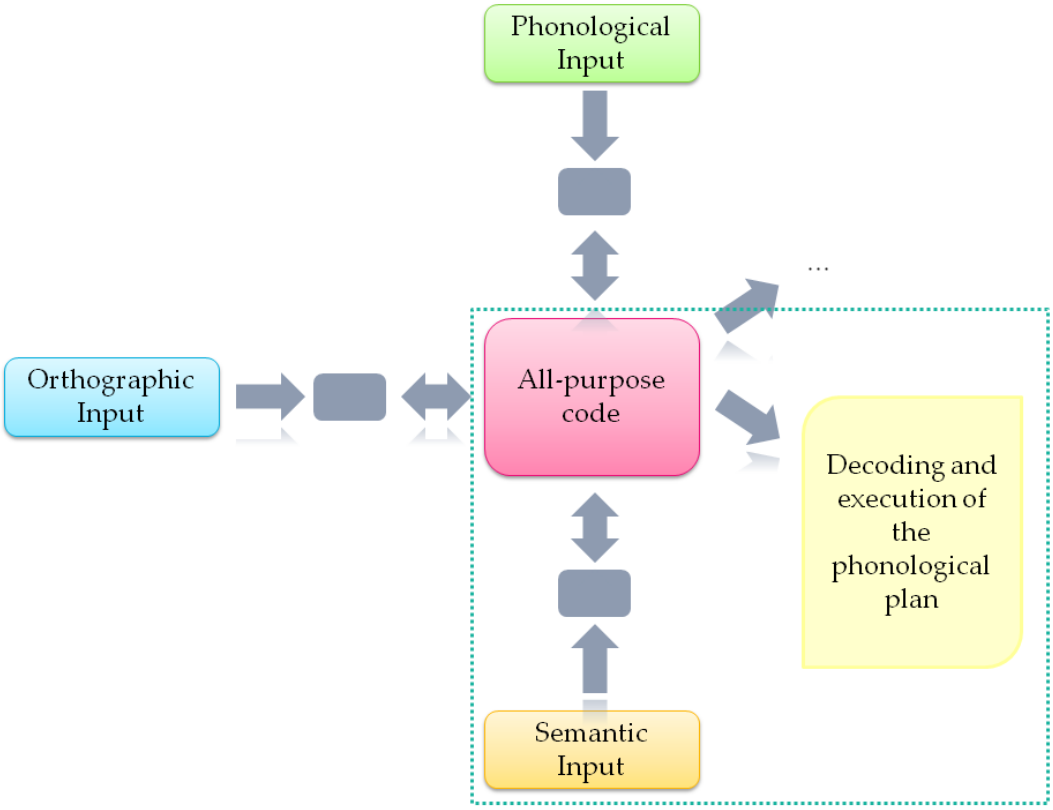


Figure 6.1: The general working framework for our modelling work. Three sources of information (orthographic, phonological, and semantics) concur to build a multiple-purpose code for words. Such code can then drive different output systems depending on the required task: in the case of reading, or production, the output system takes care of producing the sequence of phonemes constituting the pronunciation of the word. The dotted box shows the part of this framework that was actually implemented.

irregularity might not require the accumulation of further evidence. One can think of this encoding process as activating all competing codes that are compatible with the portion of the input that has been disclosed to the network up to that moment: as more information is given, previously compatible codes get discarded, and therefore ambiguity is reduced (cf. the Cohort model: Marslen-Wilson, 1987). In this view, it is straightforward to see why a later irregularity is less harmful than earlier ones.

The proposed working framework has therefore, in principle, the potentiality to account for serial effect, while remaining a parallel model. This is in fact an evolution of ideas already present in the first formulations of PDP models of reading: *"Thus, activation would begin to build up first at the orthographic units, propagating continuously from there to the hidden and phonological units and from there to the motor system in which a response would be triggered when the articulatory-motor representation became sufficiently differentiated"* (Seidenberg and McClelland, 1989, p.529).

The key feature that the model should be endowed with, then, is the ability to decide by

itself when a pronunciation can be safely started. To accomplish this, we experimented with different implementations that will be discussed in the next sections. The main idea shared by these implementations is that of conceptually distinguishing between a more cognitive part of the model, and the motor execution part. This distinction gives rise to a *two-component model*: the cognitive part of the model is responsible for the computation of an internal representation for the input word as a distributed pattern of activation; the articulatory portion of the network transforms this code into the appropriate sequence of phonemes that constitute the pronunciation of the word. An important remark here concerns the nature of interaction between the two parts: information flows from the cognitive part to the articulatory part in a *cascaded*, rather than *staged*, fashion. This means that the motor execution of the phonemic sequence associated with the internal representation for a word does not occur only after this is has been completely computed; the code is allowed to change throughout the execution of the articulatory trajectory, so that additional information coming from the input can impact the current utterance while this is in execution. Note that this is a crucial assumption in that, if staged processing was assumed, a position of irregularity effect would not be expected, or at least would be strongly reduced: the delay in the rise of a clear code, caused by uncertainty in the first intervals of input processing (corresponding to early irregularities), would be flattened in a staged processing version, as the complete code for the input word would be passed to the articulatory system after all irregularities have been resolved, independent of their position inside the string. Variations on the mechanism that regulates the flow of information between the two components constitute different implementations of our model, and will be illustrated in the remainder of this chapter.

Simplifications are often necessary when designing a new model. In this case, we first decided to postpone decisions about how to represent the orthographic input (fully sequential? fully parallel? a mix of the two?). We therefore chose to start by simulating *production* rather than reading: that is, the input to the network is a (distributed) representation of the semantics of the word. This provides a starting point for investigating the properties of the model: if it exhibits the desired behavior in this simplified setting, then orthographic input can be added to complete the model.

For all simulations the LENS software was used (Rohde, 1999): besides exploiting the built-in features of this software, we also worked on modifications of the C source code to implement the required mechanisms.

6.3 Related work

Part of the inspiration for our two-component model came from the work by St. John and McClelland (1990), who described a connectionist network trained to comprehend simple sentences. This network is organized in two parts: in the first part the constituents of the input sentence are sequentially fed into the first hidden layer, together with a copy of the previous activation values over that same layer. This hidden layer encodes knowledge about

the input sentence based on what it has seen up to that moment; this evolving representation is named *sentence gestalt*. This also becomes the input for the second portion of the model: a second hidden layer takes the current sentence gestalt and a probe (this can be thought of as a question, like "who is the agent in this sentence?" – probing for a filler –, or "what role does *the dog* play in this sentence?" – probing for a role), and projects to the output layer where an answer to the probe must be given. The training regimen forces the network to try and guess all the role/filler pairs of the input sentence at each step, thus well before the sentence has been completely processed, in order to maximize knowledge extraction across the training dataset. During the first processing steps the predictions of the network are necessarily imprecise, as many combinations of roles and fillers are possible; however *"[a]s more constituents are processed, additional evidence more strongly supports fewer possible events"* (St. John and McClelland, 1990, p.224).

This idea of an internal representation that gets refined as more information is provided is also at the basis of a model of word comprehension and production by Plaut and Kello (1999). The core of the model is the phonological (hidden) layer that supports both comprehension (activation of word meaning) from incoming acoustic input, and production (activation of articulatory output) from word meaning. Phonological representations are trained so that a semantic output can be given as fast as possible, on the basis of the partial sequence of acoustic input, *"as soon as the word can be reliably distinguished from all other words"* (Plaut and Kello, 1999). The learned phonological plan is also used to produce the correct sequence of articulatory acts. As in (St. John and McClelland, 1990) a simple recurrent network is used; notice that in both cases, as sequential information must be encoded (either constituents of a sentence, or articulatory acts in an utterance), the network must be endowed with some memory – and a SRN is the simplest architecture to achieve so.

The existence of two conceptually different parts in a model requires that a choice about the modality of interaction between them, either cascaded or staged, be made. The difference between staged and cascaded processing with respect to the relationship between cognition and action has been addressed in (Kello et al., 2000). This work reports the results of behavioral experiments employing the Stroop task[1] (Stroop, 1935) that are interpreted as evidence that cascaded and staged processing are both possible, depending on task demands. In other words, if no pressure is made for a subject to give a quick response, then staged processing can be assumed, as the internal representation can be fully computed before being submitted to the articulatory system. If, however, time constraints are introduced, the developing internal representation may be used to initiate a response, and subsequent changes in such representation can produce deviations in the trajectory – behaviorally, this is observed in the form of longer durations for utterances in the incongruent condition (that is, when the background color and the printed word denote different colors), as if the articulatory movement

[1] The Stroop effect is defined in the context of naming colors of printed words. When a subject is asked to name the color of a word, which is itself the name of a color, longer latencies are observed when the word is printed in a different color than the one identified by the word itself (e.g. the word YELLOW printed in red letters). This effect is explained as the result of interference between inconsistent semantic and color information.

was slowed down in order to allow for additional cognitive processing that could clarify the internal representation. Kello et al. (2000) also described a connectionist model that simulated the behavioral results: this is a fully continuous recurrent network with localist representations at the input layer (one unit is on for each possible color), three hidden layers with autofeedback connections, and an output layer where the naming response is given by activating the unit corresponding to the presented color. Interference from the lexical stimulus is modelled through the activation of an additional input unit. Latency and duration for a response are measured by imposing an onset and an offset threshold over activations in the output layer: when an output unit hits the first threshold, the response starts (the corresponding tick of computation is taken to be the latency); when the second threshold is crossed, the response is complete (the number of ticks between response onset and completion is used as response duration). Time pressure is controlled by the gain parameter (γ) in the hidden layers. The increase in gain determines a faster rise in activations, as $y_i(t) = 1/(1 + \exp(-net_i(t)\gamma))$, and, thus, an earlier passing of the response threshold. The simulations confirmed that longer durations were associated with the incongruent condition paired with high gain (corresponding to time pressure). Notice that it was advanced that the cascaded mode may be based on parallel encoding, in that the partial internal code that drives a response is not necessarily to be interpreted as the representation of an initial subset of the stimulus (as the first phoneme of a word), but may rather be a representation of the input as a whole, which is only approximated at earlier times but gets improved with further processing.

Time pressure can be thus imposed by the environment (e.g. in a time-constrained experiment); it can also be self-induced, as part of a control strategy chosen by the subject. It has been proposed that this can be explained in terms of the setting of a *time criterion* (Ollman and Billington, 1972): a response must be given by the time a deadline is reached, even in presence of incomplete information. This typically enforces some degree of speed-accuracy tradeoff (Pachella and Pew, 1968): the more the time allowed, the more accurate the performance. In (Kello and Plaut, 2000) this tradeoff was demonstrated on the existent computational models of reading by testing the performance of both the PMSP and the DRC model when naming was forced to occur at earlier times than normal processing. As expected, the number of errors increased when the time threshold for a response was lowered. However, the authors reported experimental evidence against a pure time criterion, advancing that a gain modulation mechanism like the one implemented in the Stroop model in (Kello et al., 2000) is more consistent with the data, as it allows for durations to be coupled with latencies; this is a consequence of the fact that all processing (not just the response onset) is subject to acceleration when the gain is increased.

The effects of input gain as a mechanism for strategic control in reading were investigated in (Kello and Plaut, 2003). To this end, the authors developed an extension of the PMSP model, endowed with an implemented semantic layer, so that both the hypothesized pathways from print to sound (the direct, phonological one, and the mediated, semantic one) are now in place. The orthographic, phonological, and semantic layer are all connected to each

other via a dedicated layer of hidden units. Connections on the "spoken word pathway" of the network are bidirectional, whereas the rest of them are feed-forward only, for the sake of simplicity. Moreover, both the phonological and the semantic layer are connected bidirectionally to their own set of clean-up units that help forming attractors. Simulations on this architecture confirmed the viability of input gain modulation as a reasonable mechanism for strategic control, for instance for implementing route emphasis (whereby the processing of one route – in the triangle models case, the semantic-mediated pathway vs. the phonological pathway – is enhanced at the expense of the other route). Without entering into details about these simulations, it is worth mentioning that discrepancies between error patterns in the above described model and those reported in (Kello and Plaut, 2000) for subjects motivated the introduction of an alternative network architecture: here, the orthographic layer maps, through a single layer of hidden units, directly to the hidden layer located between the semantic and the phonological portion of the network. This architecture allows greater integration in the operation of the two pathways, and in fact is named *integrated-pathway architecture*. Interestingly, the central hidden layer can be seen as encoding an all-purpose code such as the one we assume in our general framework.

A complete model based on the idea of an all-purpose code is the *junction model* by Kello (2006). The model is composed of three sets of 400 units each, respectively for coding orthographic, phonological, and semantic representations. These three layers are all connected bidirectionally to a core set of 45,263 units that hosts word representations. Phonological and orthographic representations were learned through two dedicated encoder-decoder systems; these are not part of the junction model itself, but rather provide it with appropriate input representations that are learned rather than engineered. In the orthographic encoding-decoding system, the sequence of letters constituting a word, coded by a pattern of visual features (e.g. vertical and horizontal lines, curves, etc.), are mapped to a distributed representation that allows for the input sequence to be reproduced in output (by the decoder part). Similarly, the phonological system performs the encoding-decoding task on word phonology, coded by a pattern of phonemic features, among which those representing lexical stress (in this way the junction model is able to deal with multisyllabic items). In (Kello, 2006) the encoding-decoding task was performed in two stages, mapping letters (or phonemes) to syllables first, and then syllables to words, but subsequent work showed that similar results can be obtained with a single stage only (Sibley et al., 2008). Semantic representations were based on co-occurrence statistics of words as computed via the COALS method (Rohde et al., Submitted).

The lexical representations at the core of the junction model are localist units (there are as many units as words in the training dataset) and are not subject to learning, for reasons of computational complexity. Rather, weights and parameters are hand-coded. This work was preliminary in many respects, for instance because performance on pseudowords was not tested; and the hand-coded, localist approach appears to be far from typical requirements for PDP models. However, it showed that the junction architecture is viable, and capable, in

principle, of explaining a large proportion of variance in naming latencies.

The use of an integrated-pathway architecture, like the one we have assumed for our work, may be seen as problematic in the light of the well-known double dissociations associated with specific forms of acquired dyslexia: these are naturally explained in terms of selective damage to either of two pathways, whereas the other is relatively spared. In an integrated-pathway architecture, this approach cannot be taken. However, it was shown (Kello, 2003; Kello et al., 2005; Sibley and Kello, 2005) that double dissociations in the reading of exceptional words and pseudowords can emerge in such unified architecture when the input gain parameter is manipulated; this was shown both in models employing distributed representations, and in localist models. By setting a low gain for hidden units, a mapping more biased toward regular correspondences was obtained; conversely, high gain corresponded to higher sensitivity to item-based mappings, with scarce generalization to novel items[2]. Modulation of the input gain parameter offers therefore a new, interesting account of how observed double dissociations might occur in a non-modular (or, at least, less modular than how it is postulated by dual-route theories) processing system.

Sibley et al. (2008) showed how two simple recurrent networks can be coupled to perform an auto-encoding task on sequences of letters, or phonemes, so that the learned internal representation acts as a (either orthographic or phonological) *wordform*. This is done by simply running the encoding SRN to the end of the input sequence, then copying activations at the final layer to the first layer of the decoding SRN, and training this network to produce in output the same input sequence. Weight updates computed by gradient descent are then backpropagated to train both the decoder and the encoder network. Wordform representations learned through the *sequence encoder* (where a wordform is represented by the activation pattern over the last layer of the encoder network) have the advantage, with respect to traditional slot-based representations, of being of fixed width, therefore solving problems of input-output alignment and dispersion. The authors showed that the method is suitable for large-sized, multisyllabic lexicons, and therefore proposed that it could be used to scale up the existent computational models of reading. Such possibility was explored[3] in (Sibley et al., 2010). Here a sequence encoder was used to map a sequence of orthographic representations (the written word) into a sequence of phonological representations (the pronounced word); the intermediate representation can be seen as a static plan that drives the execution of the phonological sequence in output. A dataset of over 14,000 monosyllabic and bisyllabic words was used, and the network showed good performances in both word and pseudoword reading. As all output sequences were produced at the same time step, a measure of naming latency was de-

[2]Manipulations of the input gain of the intermediate units has the effect of modulating the *scope of similarity* on the processing of a given item: a wider scope (low gain) means that a larger set of stimuli is processed in a similar way (in this case, in a regular way), whereas a narrow scope (high gain) causes a smaller number of items (possibly, only one) to influence a given mapping.

[3]We have seen that the junction model by Kello (2006) also employed wordform representations learned through sequence encoder networks. However, the mapping between an orthographic wordform and the corresponding phonological wordform was not learned, but rather hand-coded by direct manipulation of weights and other parameters.

rived based on mean activation values over output units, with the rationale that a higher level of activation would correspond to higher confidence in the response, and therefore to shorter latencies. Regression analyses showed that this network was able to account for a proportion of variance close to that reported for human latencies relative to a set of well-established psycholinguistic effects.

It appears, thus, that the junction model assumption (one all-purpose word code), coupled with the sequence encoder method to turn sequences into static representations, is very promising with respect to provide a sound model of reading that improves on predecessors. It should be remarked, however, that semantics still has to be integrated in this picture. Moreover, the sequence encoder mechanism retains, to some degree, an engineered, unnatural flavor, due to the process of copying activations to context units, and representations to and from the encoding and decoding parts of the network. In fact, this derives from the use of simple recurrent networks. However, SRNs can be seen as simplifications of fully recurrent networks, and are often used in their place because of the reduced computational load required to train them. It would thus be interesting to develop the fully recurrent equivalent to the system described in (Sibley et al., 2010): if the computational load can be kept manageable, such network should be able to replicate the behavior of the sequence encoder network while providing a more elegant account of the encoding process. Also notice that in a continuous recurrent network activation values rise in time, and settling criteria can be used to more naturally derive an analog to naming latencies.

In fact, our framework offers, in principle, the possibility to extend the work by Sibley et al. (2010) in the above sense, as it is a fully continuous recurrent network where orthographic input and phonological output representations are taken to be sequential, and a distributed, fixed-width code is learned for mapping the former into the latter. Although the network is conceptually organized in two sections, it is nonetheless one network, and as such the encoding and decoding processes are performed and trained as a whole. As mentioned, we have not implemented the orthographic side of the model yet, and therefore a direct comparison with the sequence encoder model proposed in (Sibley et al., 2010) is not possible at the current state of the work.

6.4 The two-component model

6.4.1 General architecture and implementation details

We can now go back to our model to provide a panoramic description of its main features. The architecture of the implemented model is depicted in Figure 6.2.

The network is conceptually organized in two parts: a cognitive part that maps the input into a static, distributed internal representation, and a motor execution part that turns this representation into a sequence of phonemes constituting the pronunciation of the word. All units in the network were initialized to be basically silent ($initOutput = 0.05$) before input presentation: the correct phonological code is expected to rise from this silence condition

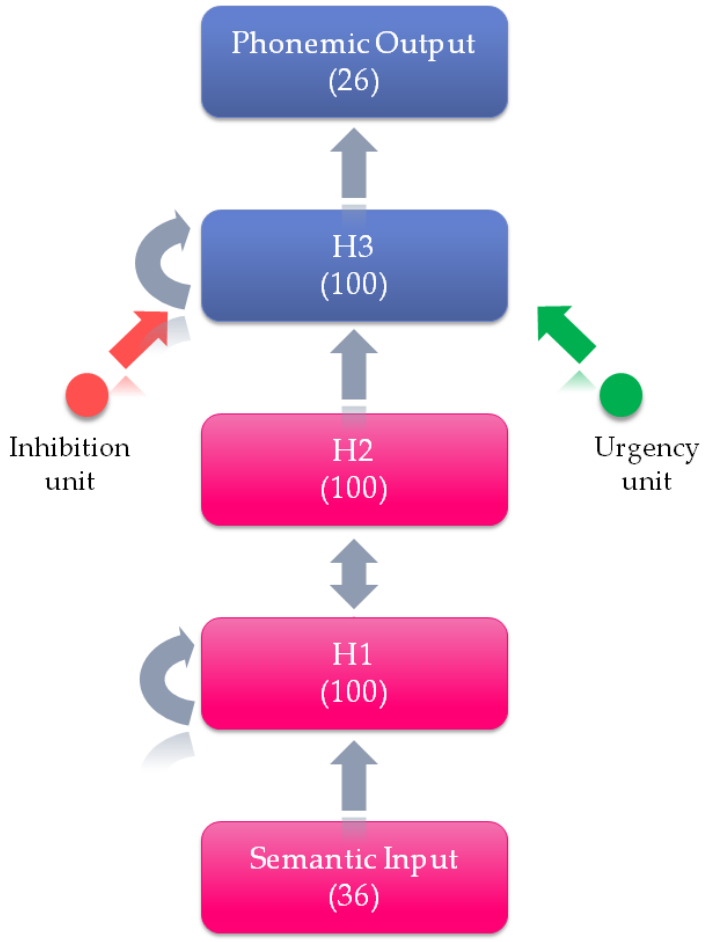


Figure 6.2: The two-component model. In magenta, the cognitive part of the model, that maps the distributed semantic input representation into an internal code (represented by unit activations over layer $H2$). In blue, the articulatory part of the network, that turns the code on $H2$ into the sequence of phoneme constituting the pronunciation of the input word. Layer sizes are reported within parentheses. The red unit stands for the inhibition unit and the green unit for the urgency unit; they both send connections to layer $H3$, the former with frozen weights -10, the latter with frozen weights $+10$. When the urgency unit is turned on, inhibition over $H3$ is released and a response can be initiated.

and get clearer as processing extends in time. Weights were initialized to small values (mean 0.0, range 0.3) for most of the network; large negative weights (mean -2.0, range 0.1) were assigned to connections from the bias unit[4] to all the other units, in order to enforce the silence condition when no input has been presented yet. All units used the logistic function as activation function.

A threshold mechanism regulates the flow of information between the cognitive and articulatory part of the model. Different variants for this mechanism have been tried, and will be discussed in detail later. All variants are based on the idea of evaluating the quality of the code computed by the cognitive portion of the network: when the code is "clear enough", that is, a threshold on the quality of the code is reached, the articulatory part of the network is signaled to start producing the sequence of phonemes dictated by the current code provided by the cognitive component.

The cognitive part of the model (shown in magenta in the figure) is composed of an input layer (36 units) and two hidden units, $H1$ and $H2$, both with 100 units. This network is a fully recurrent, continuous one: apart from the usual all-to-all feed-forward connections, feedback connections exist between layer $H2$ and layer $H1$, as well as loop connections on layer $H1$. A combination of features was introduced at the input layer in order to implement noisy activations that rise up from initial low activity, so as to simulate a gradual process of input acquisition rather than assuming that a perfectly clear input representation is immediately available to the system. To do so, activations in the input layer are soft-clamped: the activation of an input unit is given by

$$initOutput + clampStrength \cdot (externalInput - initOutput) \tag{6.1}$$

where $clampStrength = 0.9$ is a parameter used to weight the contribution of the external input and the initial value of activation for the unit. Input is also made noisy by adding random gaussian noise (mean 0.0, range 0.2). Note that the addition of random noise at the input level made our model non-deterministic. Finally, gradualness in input activation was achieved by continuous integration in time, with a time constant $dt = 0.2$:

$$newInput = lastInput + dt(newInput - lastInput) \tag{6.2}$$

The hidden units in both layers, on the other hand, used output, rather than input, integration (see Eq. 4.27); the integration constant is set to 0.5 (in other words, we have two ticks for each time interval). In general, output integration has a less abrupt effect on unit activations as output values are bounded in $[0, 1]$ whereas input values are not; the use of output integration should therefore allow for a rise in activation that is less susceptible to large fluctuations that may hamper the stability of the network.

The articulatory part of the network is a fully recurrent network that is placed, in a seam-

[4]Recall from Chapter 4, p. 101, that activation thresholds can be implemented by having an input unit, sometimes called a bias unit, clamped at 1 and connected with learnable weights to all other units.

less way, on top of the cognitive component. Recall in fact that the two-component subdivision in this model is only conceptual, since the network is one. In this articulatory part, shown in blue in Fig. 6.2, we have a hidden layer ($H3$, 100 units) that receives connections from $H2$ and from itself, and an output layer of 26 units, each representing a phoneme. Notice thus that information flows from the cognitive component to the articulatory one unidirectionally. In order to make the learning of the phonemic sequences easier, computation was made discrete in this component (that is, the integration constant is 1).

The interaction between the two components is controlled by two units that project to $H3$. The *inhibition unit* is constantly on and sends connections to all units in $H3$, with frozen weight (experimentally) set to -10. In this way, continual inhibition is exerted over the articulatory part of the network: even though this part is always observing the code computed by the cognitive component, it is prevented from starting producing the output until "it is safe to do so" – we will get back later to the exact meaning of this expression. The articulatory part of the model can start a pronunciation when inhibition is released: this occurs when the *urgency unit* (so called because it can be seen as signaling the urgency to produce a response) is turned on. This unit sends connections to all units in $H3$ with frozen weights of +10; in other words, when this unit is on, it perfectly balances the effect of the inhibition unit, so that inhibition on $H3$ is actually released. At this point, the articulatory component starts producing the sequence of phonemes for the input word.

The training dataset was the one used in (Kello and Plaut, 2003). It is constituted of 470 words that were sampled from a larger corpus of 2,802 monosyllabic words by selecting the 12 most frequent onset, vowel, and consonant orthographic clusters. Moreover, it was ensured that the distribution of alternative pronunciations for each cluster inside the dataset and the full corpus were matched, so that the same degree of consistency of the original corpus was reproduced in the sample. Frequencies to be used in simulations were computed by ranking the words in the dataset by decreasing frequencies, and applying the formula

$$F_w = \frac{1}{\sqrt{r_w + 2}}, \tag{6.3}$$

in the spirit of Zipf's law, according to which the frequency of each word w in a corpus, F_w, is inversely proportional to its rank r_w. Phonological representations are sequences of phonemes (for a total of 26 phonemes), whereas semantic representations are distributed patterns of activation that are essentially arbitrary, in that they do not code for any semantic feature and have no relation to the actual word meaning.

As our model is, as yet, a production model, input patterns represent the semantics of a word, and target patterns are sequences of phonemes (exactly one phoneme unit must be on for each subsequent interval, the others being silent) that make up the pronunciation for that word. For each example, the network is trained for 15 time intervals (enough to accommodate the longest phoneme sequence, taking into account that the initiation of pronunciation can be delayed for some intervals). The input representation is presented at the network at the first

time interval, and kept on for the duration of the trial. The sequence of phonemes is given as a sequence of targets for subsequent time steps, followed by "null events" where no phonemes must be produced (all phonemic units must be off).

However, it is not expected that the network starts producing the output sequence as soon as the input is given. As we explained above, a response is initiated only after the urgency unit has been turned on. Targets, therefore, must be assigned accordingly, only after inhibition on $H3$ has been released[5].

The network used cross-entropy (see Eq. 4.17) as the error function, and was trained through the continuous version of backpropagation through time (Pearlmutter, 1989). We set the learning rate to 0.05 and momentum to 0.8; also, a modification of momentum descent was used that is implemented in LENS under the name of Doug's momentum, that consists of scaling the weight update vector so that its norm is not greater than 1.0 – this helps reduce oscillating behavior during learning as weights cannot change dramatically in a single step. To allow for some tolerance in target approximation, a target radius of 0.05 was used. This means that if the activation for a unit is within 0.05 from its target, it is considered to be correct and no error is produced for that unit; if it is not, the target value used for error computation is set to be closer (by 0.05) to the current activation value. As each word in the training set is provided with a frequency value, these values were used during training to scale the error computed by the backpropagation algorithm: this has the effect of amplifying learning for the most frequent items. The network is trained until an error criterion is met: specifically, the activation value of each unit in the output layer must be within $trainGroupCrit = 0.1$ of its target for each example in a training batch.

We mentioned that inhibition on $H3$ is released when the code computed by the cognitive portion of the model is sufficiently clear to start a response. As processing extends in time, a code starts emerging from the noisy activity of the network. When its quality is good enough – that is, when a measure of quality crosses a response threshold – the inhibition on the articulatory system is canceled out by the urgency unit, and a response is initiated. As a consequence, as different items can require different lapses of time before their codes reach the response threshold, their pronunciation will be started at different times; this gives rise to a natural analog to human latencies in terms of ticks required to start a response.

The process whereby the internal code gets clearer as processing goes on is here referred to as a process of *evidence accumulation*. In decision making theories, this expression is used to denote the process of collecting information based on which a choice is to be made: for instance, in lexical decision information on the "word-ness" of the input stimulus can be thought of as growing in time as the input is scanned. If the evidence is strong enough, a positive decision can be made (i.e., a yes response to the lexical decision task is given). Information accumulation models, such as, for instance, the diffusion model (Ratcliff, 1978, 1980, 1981, 1988), assume that a decision is made when the evidence accumulated over time

[5]To accomplish this, an additional service layer, having no connections with the network described here, was introduced for the only purpose to store targets, so that they could be copied to the respective output units at the correct time intervals.

hits a threshold. Our approach is similar: when the quality of the code being computed by the cognitive part of the model reaches a threshold (i.e., enough evidence has been collected from the input for a response to be reliably based on such evidence), a response can be initiated.

In order to provide a complete model, two aspects of this evidence accumulation mechanism must be made explicit: the specific quality measure for the internal code (what makes a code good?) and the value, and setting process, for the response threshold. We will first provide a description of the quality measures we adopted, and then different ways to set the response threshold will be illustrated.

Before detailing these aspects, it is useful to sum up the main features characterizing the presented model:

- the model is based on a two-component architecture that links a cognitive part, which builds an internal code for the input stimulus, to the articulatory part, which is responsible for producing sequential output (the pronunciation of the word);
- the cognitive part of the model computes an all-purpose code (that is, a code that can be used to perform different language tasks, such as reading, production, comprehension, etc.), which makes this model truly single-route;
- a cascaded mode of operation between the two parts is assumed;
- an evidence accumulation mechanism works at the interface between the two parts, whose role consists in releasing inhibition over the articulatory part when a quality threshold is crossed by the all-purpose code.

6.4.2 Measuring code quality: different approaches

Average activity on $H2$: the utility unit

The cognitive portion of our model computes an internal code for the input word in the form of a pattern of activation over layer $H2$. Since we expect a code to emerge from the initial, noisy low-level activity, and get stronger as more information flows in, it makes sense to formalize our quality measure based on the average activation of $H2$. The computation of such measure can be carried out by a dedicated unit that we named *utility unit* as it, in fact, computes the utility of the current code in producing a correct output. The utility unit receives connections from every unit in $H2$, each with frozen weight $1/|H2|$; because the utility unit is set to be a linear unit, its activation corresponds exactly to the average activation level of layer $H2$. When the information accumulated by the utility unit hits the response threshold, the activation of the urgency unit is set to 1 so that the output can start to be produced.

Measures defined on the net input to $H3$

We reasoned that information about the code being built by the network does not necessarily lie only on the activation level of $H2$. Actually, in neural networks a large amount of information is coded in the connection weights that are learned in performing a given task. Thus, we

considered that a measure that combined both activations on $H2$ and weights of the connections linking the two parts of the model would be more informative. Measures based on the net input[6] to layer $H3$ were therefore conceived.

We experimented with different net input-based measures: average net input (A-net), average positive net input (AP-net, only positive contributions enter the computation), and average absolute net input (AA-net, contributions are taken in absolute value before averaging them). These modifications were introduced to take into account the fact that in a simple average, positive and negative contributions mutually cancel out. In the first case (AP-net), negative contributions are ignored, because they have an inhibitory effect on respective units. In the second case (AA-net), the sign of each contribution is neglected as only its size is considered to be informative: a strong net input, whatever its direction (positive or negative), is taken to be a sign of a clear, strong underlying code.

The measures described above are *absolute measures*. However, we also tried a *relative measure*, whereby a good code is one that is stable, that is, it is not changing much over time: $|\text{A-net}(t) - \text{A-net}(t-1)|$. In this case, a good code is associated with a small value of such relative quality measure, meaning that the information in the code has mostly settled.[7]. Notice that when using net input-based measures, the utility unit is not implemented.

6.4.3 Setting the response threshold

A related, but mostly independent issue with respect to the definition of a proper quality measure for the computed code consists in defining how the response threshold is set. Note that we are specifically referring to the handling of the threshold during training; in the testing phase, the "normal" threshold that has been used during training can be manually moved, to simulate speeded naming and, therefore, study how the induced accuracy-speed tradeoff impacts on the performance of the model. Also notice that it cannot be always guaranteed that the quality measure for a given code reaches threshold before the end of a training episode (especially at the beginning of training): for this reason, a strict deadline was implemented, such that, if for the current training example the response threshold has not been crossed within the allowed maximum time (in the performed simulations, $maxSettlingTime = 16$ ticks), a response must be provided in any case at $maxTime = 20$ ticks[8].

Training on a fixed threshold

One possibility consists in training on a *fixed threshold*. A value for this parameter can be experimentally found, and manually set so that the articulatory part of the model is unblocked when the measure of quality computed for the current internal code gets greater than this

[6]Note that the contributions of the inhibition and urgency units were not included in the computation of these measures, as they are fixed values not relevant for determining the quality of the computed internal code.

[7]This is a rough measure of stability, however, as it is based on the difference in average values, rather than differences over single values.

[8]The 2-interval (4-tick) difference between *maxSettlingTime* and *maxTime* was introduced to allow for information in layer $H2$ to actually reach the output layer.

value. For instance, the model can first be trained to produce all responses at *maxTime*, and the average value of quality measure for codes at *maxSettlingTime*, obtained at the end of training, can be taken as a reasonable value for the response threshold. In practice, this training process is used only to derive a threshold value, but the results of training (i.e. connection weights) are discarded. The model is then trained adopting the so-found threshold value.

Training on an adaptive threshold

The alternative to the use of a manually fixed threshold is to have the network discover a proper threshold by itself, during training, by means of an adaptive process. To this end, a two-phase learning regimen was implemented.

The first phase works as an initialization step, and may be thought of as the parallel of earlier learning in children when there is no pressure to produce a fast response, but attention is focused on accuracy. Therefore, in the first training phase, the network is trained to initiate all responses at *maxTime*; no threshold is used. This is followed by a second phase of learning that builds on the previous one (i.e., connection weights are not reset, but those learned at the end of the previous phase are used), that aims at gradually decreasing the response threshold. The initial threshold for phase 2 is set based on the quality measure values recorded over the training set at the end of phase 1. In different simulations, the maximum, or the minimum value of the measure over all examples was used. The network is then trained, using this value as the initial threshold; whenever the accuracy of the network gets greater than 90% examples correct, the current threshold is decreased (by 5% its current value), and learning goes on. Each threshold is maintained for a maximum number of training runs ($maxUpdatesPerTh = 10,000$); if the accuracy criterion is not reached before this temporal limit, we go back to the previous threshold, and train until the error criterion is met. This threshold will be the final one. This training regimen therefore implements subsequent adaptive steps: if the network is successful on a given threshold, it is forced to get faster than that, but only to the point where this time pressure does not significantly impair overall accuracy.

6.5 Results and discussion

In the previous section we have outlined different implementation details that have been adopted in the various versions of our model. As it can be seen, many combinations of quality measure, training regimen, and other parameters (such as hidden layer size, stopping criterion for training, initial threshold value) are possible. Giving a detailed discussion for each possible combination is not viable here: thus, we will focus on only a subset of these versions.

Recall that our model is, as yet, a production model: as the orthography of the word is not provided to the model, and the input patterns are abstract representations of semantics, we cannot look for classical psycholinguistic effects like, for instance, the regularity effect. However, in order to assess whether our model was working as expected, we looked for fre-

quency effects, both on latencies (i.e. number of ticks elapsed before the response is initiated) and on errors[9]. The assumption here is that more frequent items will be associated with more strong, stable codes, that get resolved more quickly than those of low-frequency items; therefore, the most frequent items should be pronounced earlier, and be less prone to error when the network is forced to produce an output in speeded condition. Following (Kello and Plaut, 2003), we consider high-frequency words those having frequency value greater than 0.15, and low-frequency words those with a frequency value smaller than 0.07.

Training on a fixed threshold

Let us start with simulations where training was carried out on a fixed threshold. The three absolute quality measures (A-net, AP-net, AA-net) described above were all implemented, but since results were similar in the three cases, we will discuss only results for measure AA-net. In this simulation the response threshold was set to $th = 2.1$ and training consisted of $33,641$ runs. Accuracy at the end of training was around 100%[10].

However, no frequency effect could be found on response times; it turned out that low-frequency items even had a trend toward being pronounced faster than high-frequency words ($F(1,308) = 3.4599$, $p = 0.06383$). Thus, the network did not appear to be sensitive to frequency.

Lowering the response threshold corresponds to introducing time pressure, which in turns should translate in an increase in errors. When we tested the model with a $th = 1.9$ accuracy dropped to about 77%. We looked for a frequency effect on errors, to assess whether low-frequency words were more prone to error than high-frequency ones. In fact, accuracy was about 93% for high-frequency words, and 75% for low-frequency ones.

The lack of strong frequency effects on latencies invited to investigate the reasons of such outcome. In particular, by looking more closely at the behavior of the network, we could identify some potential problems with this approach.

The first critical point is how to choose the training threshold. In this simulation a threshold was chosen empirically, as explained above. Although it might appear that this choice is of secondary importance, it should be noticed that choosing the "wrong" threshold can in fact produce undesirable results. A too high threshold means that (nearly) all items will be just pronounced at *maxTime* (recall that when the response threshold is not reached by *maxSettlingTime* a response must be produced anyway): the threshold mechanism is basically disregarded, and the possibility of reasoning on different output times is lost. Conversely, if the threshold is too low, all items will just be pronounced as soon as the information starts arriving at layer $H2$: let us call these items *immediate pronunciations*. Consider also that it is difficult to assess accuracy when testing on a lower threshold (speeded condition),

[9] At test time, accuracy was established based on the $testGroupCrit = 0.3$ parameter; this means that an example is to be considered correct if each output unit has an activation that is within 0.3 of its target.

[10] Notice that the injection of random noise at the input level makes the model probabilistic, in that each run can result in slightly different results. For instance, in the simulation we are describing here, different test runs on the network produced accuracy percentages varying between 99% and 100%.

for both immediate pronunciations and items pronounced at *maxTime*. As for the former case, these items are just not affected by decreasing the threshold, so they are not informative. Items pronounced at *maxTime* (because they do not reach threshold) might not reach even the lowered threshold; therefore, even though they appear to be correctly pronounced, this is not technically true, as the response threshold was just ignored. Special care should therefore be taken for rating these cases as errors during the test phase.

Another problem was revealed when testing the network on a higher threshold than the one that was used during training. This is the equivalent of waiting more time than usual before giving a response (at-leisure mode). We would expect that waiting more time means having an even clearer code, and therefore the accuracy of the model should be at least equal to the one obtained with the original threshold. However, this was not the case, and degradation of performance was observed for all the quality measures we tried.

The reason for this appears to be lack of stability in the learned internal code: that is, the code seems to be optimized for producing a correct response at a given point in time (as determined by threshold crossing), but quickly degrades outside a small time window. In turns, such lack of stability might be produced by different sources; we hypothesized the following:

1. an inappropriate quality measure;
2. overfitting issues.

The first point was suggested by the observation that the absolute quality measures we adopted (A-net, AP-net, AA-net) are not monotonically increasing: some fluctuations over time do happen (as it can be seen in Fig. 6.3). For this reason, a relative quality measure, here called *stability*, was introduced (see Section 6.4.2), so that a pronunciation is started when the internal code is not appreciably changing anymore. This approach, however, suffered from another problem: the use of such a stability measure actually determined wide oscillations in the network error during training, suggesting that, contrary to our intentions, the introduction of this relative measure made the network even less stable than it was when absolute quality measures were adopted.

As for overfitting, the hypothesis was that the network might be getting too finely tuned to the learned threshold, losing any flexibility to other thresholds, so that changing the threshold in either direction resulted in disruption of the dynamics of the network. In turns, overfitting could result from an over-sized network, or from too prolonged training. As for the first point, we experimented with different hidden layer sizes to assess how this parameter affected the performance, and in particular the stability, of the network. We will describe the results of these investigations in the next paragraph.

The issue of reducing training time has not been tackled yet, but will be considered in future work. However, there is already some evidence that overtraining can indeed have a role in the problems delineated above. First of all, in a preliminary simulation on a network with 50 units in each hidden layer, using the stability measure, we observed that stopping training after 50,000 runs resulted in somewhat greater stability than when the network was

trained until the error criterion was actually met (around 64,000 training runs). In fact, when the extensively trained network was tested on the training threshold ($th = 0.02$) all items were (correctly) pronounced at tick 8, while with a $th = 0.0$ (which corresponds to giving a response at *maxTime*, since the stability measure can never go exactly to zero) the percentage of correct responses dropped to about 11%. On the other hand, testing the network that was subject to shorter training at $th = 0.02$ yielded a wider range of output times (still achieving 100% performance), and the $th = 0.0$ condition was associated with an accuracy of about 83%, revealing a general higher degree of robustness at this earlier stage in training. It should also be noted that the lack of frequency effects that was reported above might be linked to overtraining as well, as extensive exposure to the training corpus might have contributed to flatten differences in frequency to the point that all items, regardless of whether they have high or low frequency, are perfectly mastered by the network.

The effect of hidden layer size

Hidden layer size directly affects the performance of the network because it determines the computational power of the network itself: if the number of hidden units is insufficient, a difficult task might not be mastered (and mapping from arbitrary semantic representations to phoneme sequences is indeed a hard task, as it is non-systematic, as opposed to, for instance, the orthography-to-phonology mapping); on the other hand, if there are too many hidden units, overfitting might occur.

We therefore investigated whether diminishing the number of hidden units in the model could result in better performance, especially in terms of stability of the internal code. This was assessed by looking at the behavior of the network when tested on higher thresholds than the one used during training: as ideally one would want to have comparable performance to the case when the training threshold is used, higher percentages of accuracy are here taken to be a sign of a higher degree of stability. We therefore varied the number of units in each hidden layer by setting the variable *numHidden* to one of four possible values: $\{100, 75, 50, 40\}$. We then trained the four resulting versions (using, as above, the AA-net measure) on a fixed response threshold, and tested them on a set of different values of response thresholds. This process highlighted that hidden layer size also has an impact on the range of values assumed by the employed quality measure. When training the network with $numHidden = 50$ on the same response threshold used for the network with $numHidden = 100$, all items were pronounced at the same, early output time (on tick 9), showing that such threshold was too low for this network. Figure 6.3 shows the values assumed over time by the AA-net measure when three networks (with $numHidden = 100$, 75, and 50, respectively) were tested on the word TO: it can be appreciated how such values tend to increase with decreasing hidden layer size. For this reason, we trained the networks with $numHidden = 100$, 75 with a $th = 2.1$, and the networks with $numHidden = 50$, 40 with a $th = 3.0$. Figure 6.4 shows the accuracy of each of the four versions of the model when tested at different thresholds; it can be seen that in these simulations there was no clear advantage in decreasing the number of hidden units.

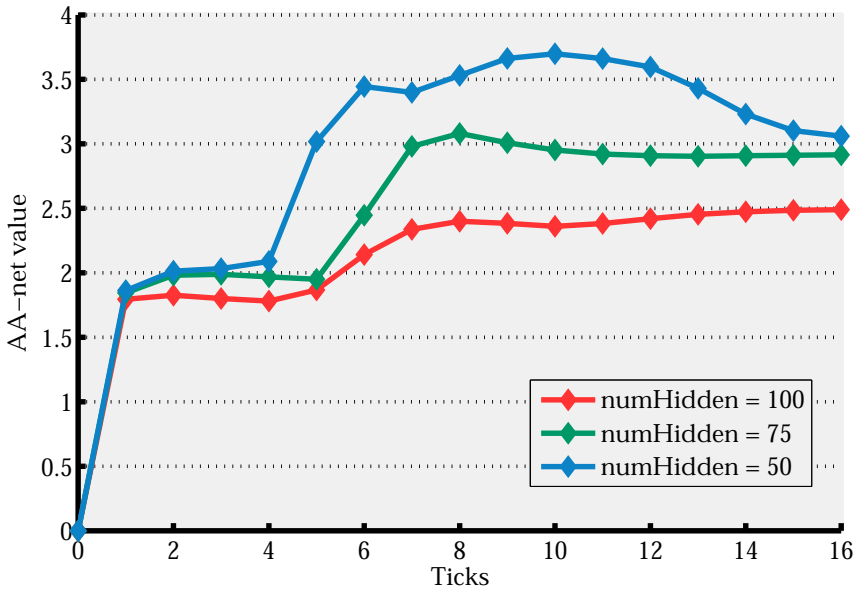


Figure 6.3: Values taken by the AA-net measure when testing three differently sized networks on the input word TO.

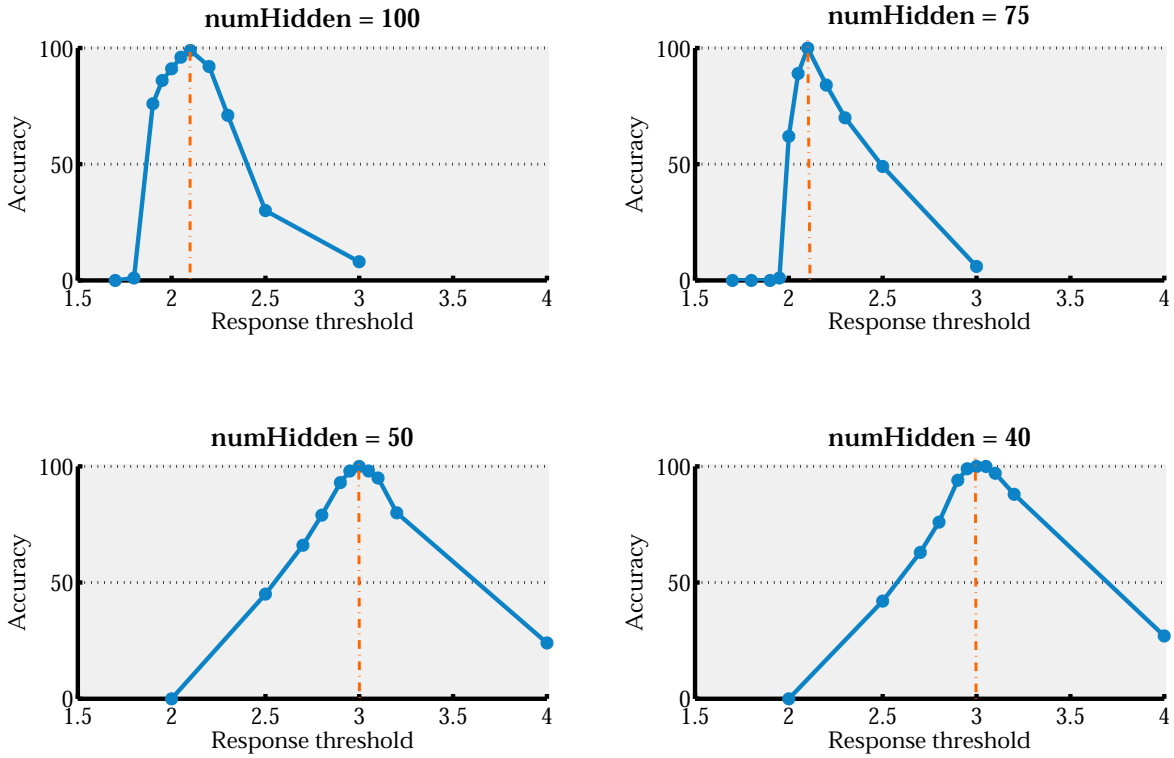


Figure 6.4: The graphs plot accuracy when testing on different thresholds, for four differently sized networks. Vertical dashed lines show the response threshold that was used during training. Ideally, the portion of the graph at the right of the vertical line would be mostly flat, with accuracy at ceiling.

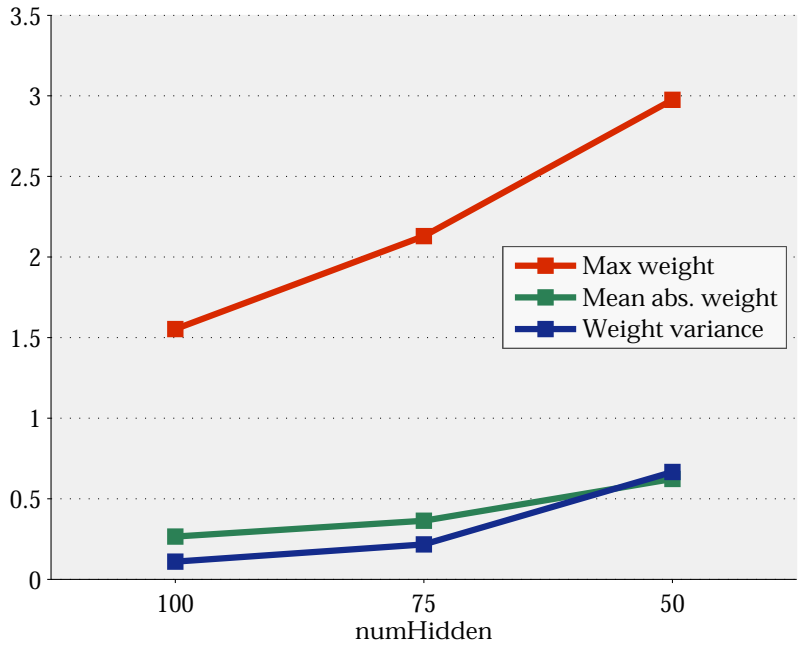


Figure 6.5: Statistics of connection weights from layer $H2$ to layer $H3$ for differently sized networks.

On the other hand, decreasing the hidden units indeed made it harder for the network to learn to task, as the number of training runs required to reach the error criterion increased with decreasing number of hidden units (33,641, 36,761, 69,408, and 86,562 training runs, respectively).

Lastly, we mention that the reason why a smaller number of hidden units is associated with larger values of the quality measure lies in that weight magnitudes for connections from layer $H2$ to $H3$ increase as well. Figure 6.5 illustrates this point, by graphing the maximum weight, the mean (absolute) weight value, and the variance of weights for connections between these two layers, for three different sizes. It appears that networks compensate for a smaller hidden layer size by growing larger weights. However, this is not a desirable feature as large weights tend to be associated with low stability (as they amplify noise).

Training on an adaptive threshold

We have seen how the implementations of the model tried so far were lacking in stability, and how changing the size of hidden layers was not decisive in improving this aspect. This problem was noticed when testing the network on higher thresholds than the one it was trained on, and observing that accuracy was degrading when it would be excepted to stay basically constant, or even improve. This observation pointed indeed to a weakness of this network; however, it might perhaps be advanced that increasing the threshold is equivalent to retain the computed code in working memory for a certain interval, so that if this time span

is too prolonged, a degradation of performance could be somewhat plausible.

In any case, achieving better stability appears to be a crucial issue here. We reasoned that a different training approach, based on an adaptively learned threshold (see Section 6.4.3) rather than a fixed one, could help with such issue: if the network is trained on a range of different thresholds, it might be expected that it is also more robust to small threshold changes, therefore possibly limiting the problem we discussed about stability and threshold increasing. Moreover, learning a threshold in an adaptive fashion looks like a more natural and elegant approach than just fixing an arbitrary threshold, and alleviates the problem of how to determine an adequate threshold value.

We therefore implemented the two-phase training paradigm described in Section 6.4.3, starting with a mid-sized network (i.e., $numHidden = 50$); the employed quality measure was again the AA-net one. The first phase of training required $60,244$ runs; at the end of this phase, the value of the quality measure for the internal code generating the phonemic response at *maxTime* was collected for each item in the dataset. The maximum among these values was used as the initial value for the response threshold ($th = 4.18304$) to be employed in the second training phase. The second phase ended at training run $88,648$ (thus, this phase lasted about $28,000$ runs) with a final $th = 1.88237$. Notice that we constrained the range of possible thresholds so that the network would stop trying to decrease its current threshold when the minimum value was reached; this was the case for this simulation. At test, all items were (correct) immediate pronunciations. Because of this, it is not possible to discuss any effect on response times. Thus, it appears that the network had no difficulty in accurately learning the task, even when repeatedly decreasing thresholds, to the point that the threshold is immediately passed. For this reason, we did not try to implement the $numHidden = 100$ version.

Even though we cannot say anything about frequency effects as all items are pronounced at the same latency, we can nonetheless look into the issue of stability to assess whether the adaptive training regimen induced a more stable behavior in the network, with respect to training on a fixed threshold. In fact, accuracy seemed to be preserved when increasing the response threshold, but only for a limited interval, after which accuracy dropped again (see Fig. 6.6). This is due to the fact that the values for the quality measure rise quickly, so that not only the final (lowest) threshold is immediately overcome, but also subsequent thresholds (in this example, up to $th \approx 3.0$); increasing the threshold up to that point, thus, does not actually change the time of threshold crossing.

The behavior of the trained network, in general, highlighted the relevance of the initialization step for the response threshold at the beginning of training phase 2. Here, we have used the maximum value assumed by the quality measure over the training set, at the end of training phase 1. As a result, in the first runs of phase 2 the majority of the training items do not even reach such threshold, and are just pronounced at *maxTime*. Notice that these events are not classified as errors; therefore, in few runs the network reaches the 90% accuracy criterion set for continuing to the next, decreased threshold, although for many items there was

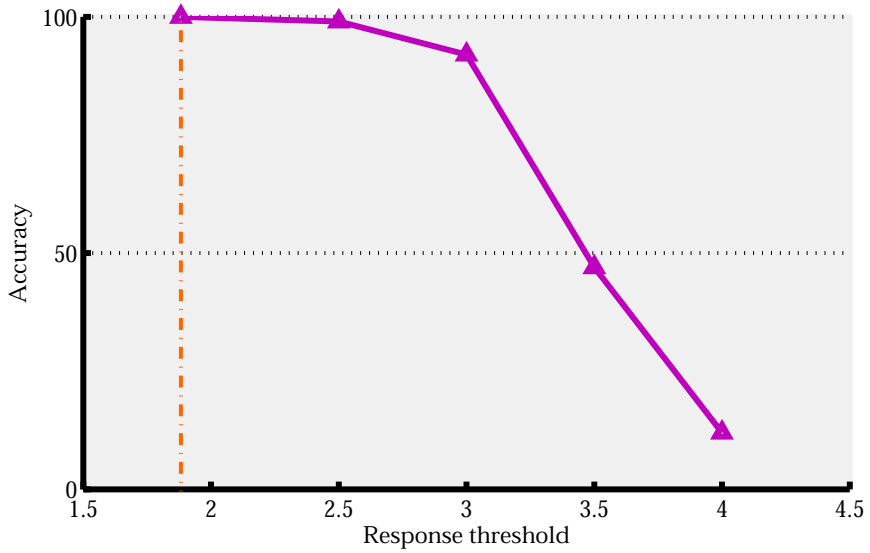


Figure 6.6: The accuracy for a $numHidden = 50$ network trained with the two-phase regimen to adaptively learn the response threshold is plotted, when tested on increasing thresholds.

no actual pressure to increase the strength of their internal representations (as no error was generated, and therefore there was no weight correction in this direction). As a consequence, the network lost adaptivity, as in fact the first thresholds were ignored, and it was directly trained on much lower threshold values. Moreover, if the $maxUpdatesPerTh$ parameter is set to a high value, the effects of training on a threshold can be easily washed out by subsequent, extensive training on a lower threshold. These observations suggest that particular care must be taken in tuning both the initial value for the response threshold, and the number of training runs during which the same threshold value is maintained. Unfortunately, there appears to be no straightforward way to "optimally" set these values. One could reason that the minimum value for the quality measure over the traning set can be used instead; but in this case, the initial threshold for training phase 2 would be far too low for the majority of items. If a low value of $maxUpdatesPerTh$ was used, the network might just not be able to reach, within such time limit, the 90% accuracy criterion, and therefore quit training; on the other hand, if $maxUpdatesPerTh$ was set to a high value, the network would be basically re-trained directly on a new, low threshold, and therefore adaptivity would be lost. An alternative would lie in setting the initial threshold for training phase 2 to the average of the quality measure values over the training set; but the issue relative to the choice of $maxUpdatesPerTh$ would remain, since, if the network was given enough training runs for each subsequent response threshold, all of them would be learned, again until the point where the current threshold would be so low that it would be exceeded from the very beginning, resulting in immediate pronunciations.

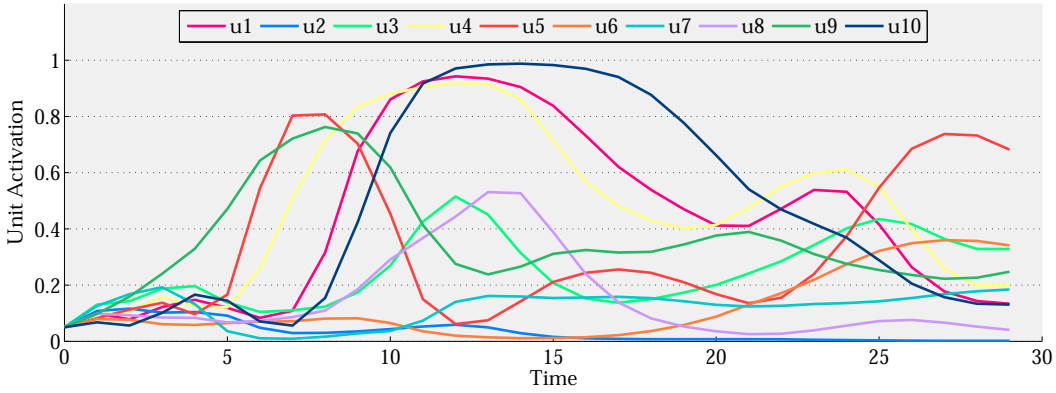
Also the adaptively trained version of the model, therefore, turned out to be dependent on the choice of the initial threshold value and on the extent of training. In fact, overtraining appears to possibly have a role here as well: training, on both phases, could be stopped at an earlier stage in an attempt to improve generalization to a larger range of thresholds.

We also tried to get a better insight into the network dynamics by looking at the behavior of single units under different conditions. In particular, we compared the activations of a set of units in response to the same input word, when adopting two different response thresholds. As we mentioned above, the network reaches perfect accuracy with the adaptively learned threshold ($th = 1.88237$), and lowering this threshold has no effect on accuracy; therefore, for this analysis we contrasted the learned threshold (from now on, th_L), to a larger one (let us call it th_F, meaning that it is a "forced", rather than learned, threshold). At $th_F = 3.0$, accuracy is still good ($\approx$ 91%) but errors do occur, as it is the case for the input word BEAD. We therefore compared unit activations in the case of correct pronunciation of BEAD – obtained when testing on th_L – and in the case of a pronunciation error – obtained when testing on th_F.

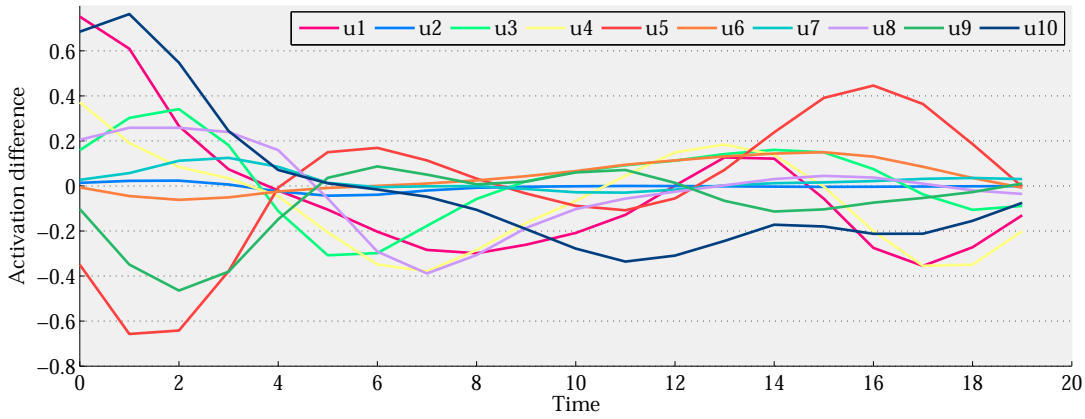
Let us start by considering layer $H2$: actually, activations on this layer are not different in the two cases, because the change in response threshold affects only later layers. What is interesting to compare here are the activation values that, in the two cases, reach the articulatory part of the network at inhibition release. This occurs at different points in time (precisely, on tick 7 and 10, for th_L and th_F respectively) depending on the test threshold. Figure 6.7(b) shows the difference in the activation level of the first ten units in $H2$, over ticks, obtained by aligning the temporal sequences for the two conditions, so that the first plotted value for each curve is $y_p(10) - y_p(7)$ (where $y_p(t)$ denotes the activation value for unit p at tick t); in a sense, the figure depicts an account of the stability of layer $H2$. We can see that large variations are concentrated in the first ticks, with a higher degree of stability emerging later on. This is confirmed when looking at the actual activation values of these ten $H2$ units over time (Fig. 6.7(a)): although units tend to have fluctuating behavior over the whole temporal sequence, they experience steeper changes at its beginning (the very first ticks should not be considered because input information reaches this layer with a delay). This observation might suggest that one should not train the network on too low thresholds, to allow for units to reach a more stable behavior. Notice also that while some units (e.g. units 1, 4, and 10) do show a desirable behavior, as they grow in activation value and then maintain it, at least for some interval[11], others (e.g. unit 5) present wide fluctuations. Such behavior might be, as discussed above, a consequence of an overtraining regime.

We then went on to consider later layers, namely $H3$, and the output layer. Figure 6.8(a) shows the values over time of the net input to the first 10 units of layer $H3$, for the testing condition th_L. The net input is consistently negative for all units until inhibition is released: at that point, while some units still continue to receive inhibitory input (in this case, unit 5), others get excitatory input, some more consistently (e.g. units 4, 8, and 10), others only transitorily (e.g. units 1 and 9). "Persistent" units might be particularly involved in keeping the

[11] Also notice that later ticks, plotted here for completeness, are not strictly relevant for pronunciation as the last phoneme of the word is pronounced on ticks 15-16 and 18-19, on condition th_L and th_F respectively.



(a)

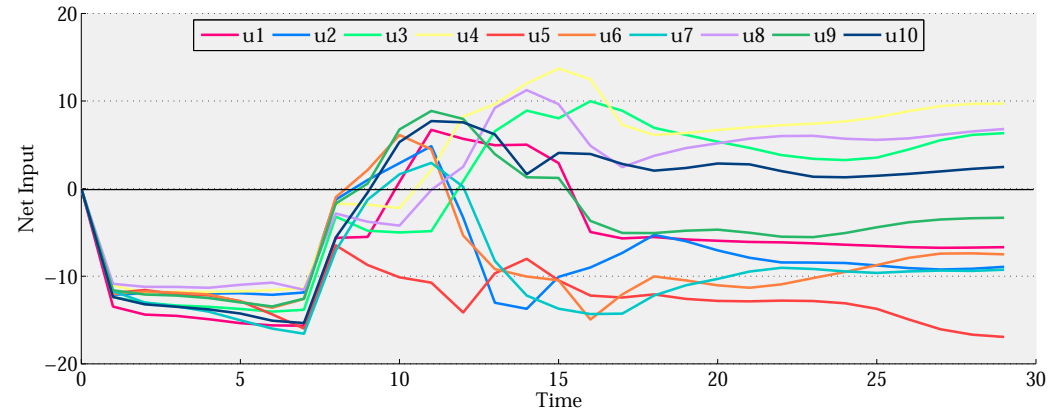


(b)

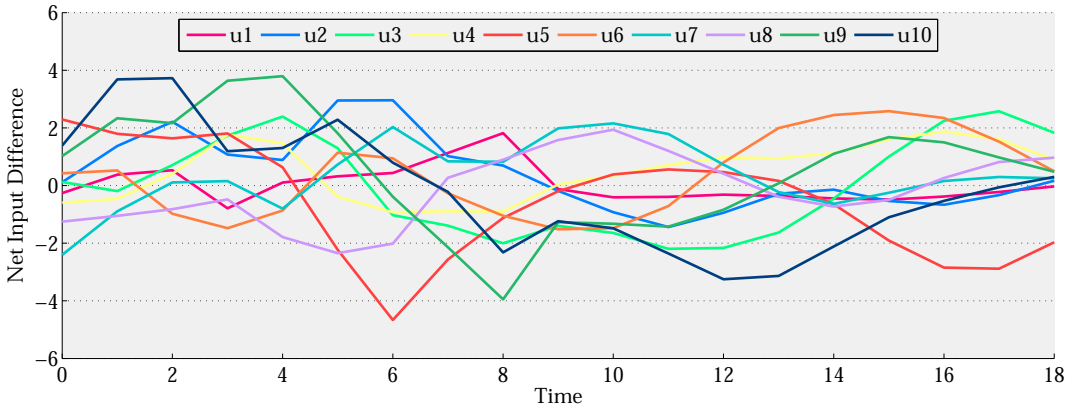
Figure 6.7: Panel (a) shows, for each tick, the activation levels of the first 10 units in layer $H2$ of a $numHidden = 50$ network trained on an adaptive threshold, when tested on the input word BEAD. The dynamics of unit activation are various, and fluctuations are often observed. In panel (b), the difference between the activation values reaching the articulatory portion of the network at the time of inhibition release (which is different for the two considered thresholds) is plotted: large differences can be observed especially at the first ticks.

output units silent after the last phoneme of the word has been produced; however, weights inspection revealed that, if any, the specialization is only partial, as these units do send excitatory connections to some units in the output layer. A comparison of the net input received by each unit, after inhibition release, under the two considered conditions, is shown in Figure 6.8(b) in the form of activation differences over time (i.e., each point in a curve is computed as $net_{th_F}(t_1 + \delta) - net_{th_L}(t_2 + \delta)$, where net_{th} denotes the net input computed under testing condition th, and t_1 and t_2 represent, respectively, the first tick after inhibition release under the two testing conditions). The main message to be taken from this graph is that the net input reaching layer $H3$ can vary to a considerable degree across testing conditions, as a result of differences in output values sent by layer $H2$ and amplified by connection weights, and of the recurrent connections existing on layer $H3$ itself. These differences, of course, can be appreciated at the output side of the layer as well: Figure 6.9(a) gives a profile of activation values for the same 10 units in $H3$, under condition th_L, where the sequential activation of some units and the persistent activity of others is seen again. As before, we also show the activation difference values obtained after realigning the temporal sequences corresponding to the two testing conditions (Fig. 6.9(c)). We can appreciate how the largest differences can be traced to those units that we previously identified as persistent (especially units 10 and 8 – with the exception of transitory unit 9). However, when looking at the activation profile for condition th_F (Fig. 6.9(b)), it can be seen that the overall behavior is not dramatically different, with variations that mostly interest the rate of increase or decrease in activation. By comparing Figs. 6.8 and 6.9 it can be appreciated how in this example the saturation quality of sigmoidal units has helped even out some of the differences in the network dynamics introduced by the change in testing threshold.

Lastly, the behavior of output units was analyzed. Figure 6.10 shows the activation level for five output units in both testing conditions; for the sake of clarity, we plotted activations only for these units as these were the only ones reaching, at least for one tick and in one condition, a minimum activation value of 0.1. Under the testing condition th_L the pronunciation of BEAD is correct, and in fact Figure 6.10(a) depicts a very clean activation profile where all the output units that do not code for any phoneme in the input word are silent. A different situation is depicted in Figure 6.10(b): the first phoneme is mistakenly taken to be /m/ (although with a rather low activation value), the second phoneme is strongly activated only for one tick out of two, and for the last phoneme, although this is correctly identified, there is also a relevant activation of another unit (corresponding to phoneme /n/). Therefore, it seems like the network was unsure whether the input word was BEAD, as it was the case, or rather MEAN. Interestingly, it turns out that the semantic representations for these two words share 5 units out of 8: therefore, we can advance that the similarity in the input, combined with the similarity localized on the output vowel /E/, resulted in similar internal representations, so that, when the network computation was perturbed (as it is the case of a change in the response threshold), the similarity of such representations possibly caused ambiguity at the output stage. Let us also notice that a mere similarity in the input structure does not appear,



(a)



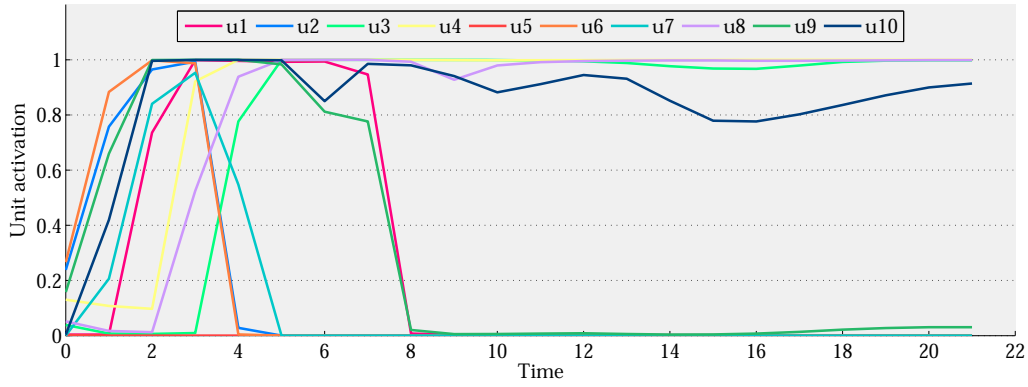
(b)

Figure 6.8: In panel (a) the net input to the first ten units of layer $H3$ is shown over time, under test condition th_L; the difference in the net input received by each unit after inhibition is released, under the two testing conditions, is graphed in panel (b).

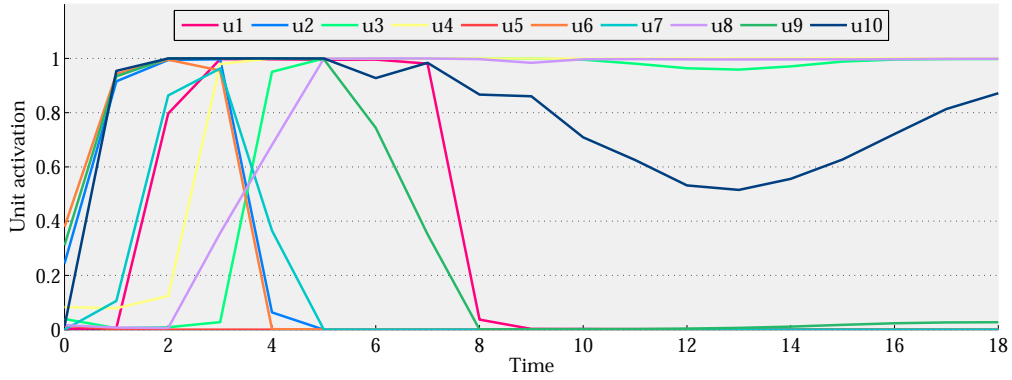
alone, to produce the same results; for instance, the input representation for the word BUCK also shares 5 units with the one for BEAD, but nonetheless none of its phonemes achieved a significant activation value.

Taken together, the analyses focused on single unit behavior confirm the general impression of a lack of stability in the neural code developed by layer $H2$, possibly due to overtraining; they also highlight the complexity of unraveling the dynamics of a recurrent neural network, which makes the process of blame assignment, when the results are not as expected, a hard, and painful one.

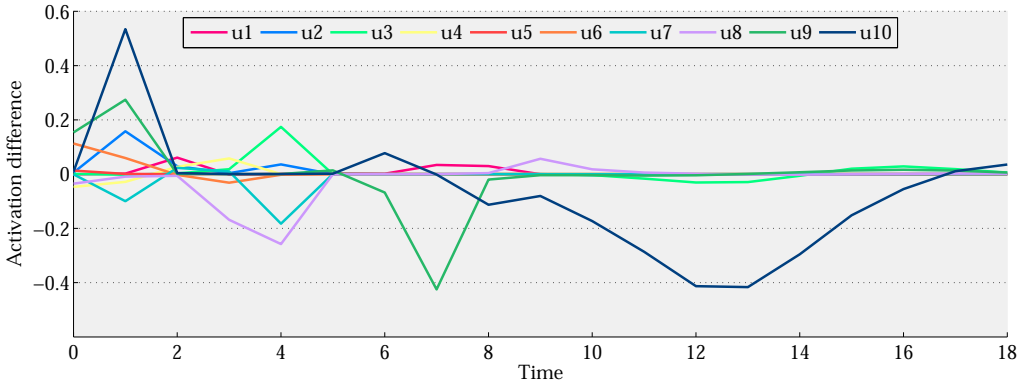
A final, alternative version of the model was targeted at addressing a potential weakness of the two-phases training regimen described so far, namely, the handling of those items that



(a)

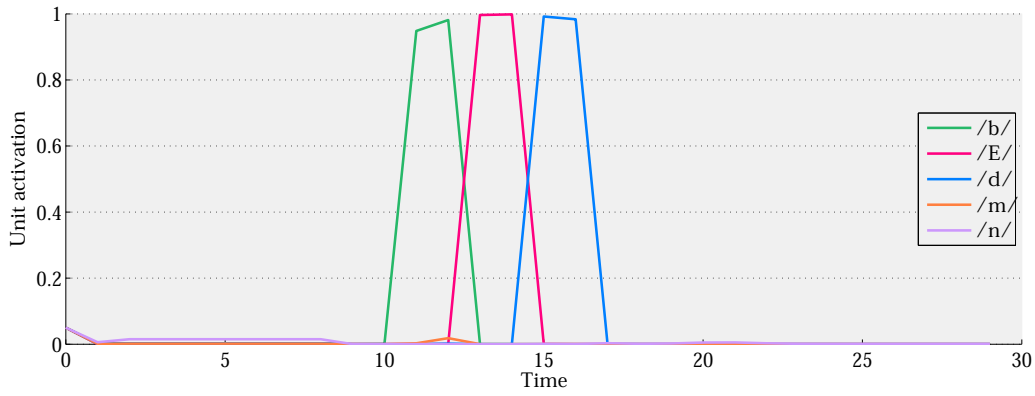


(b)

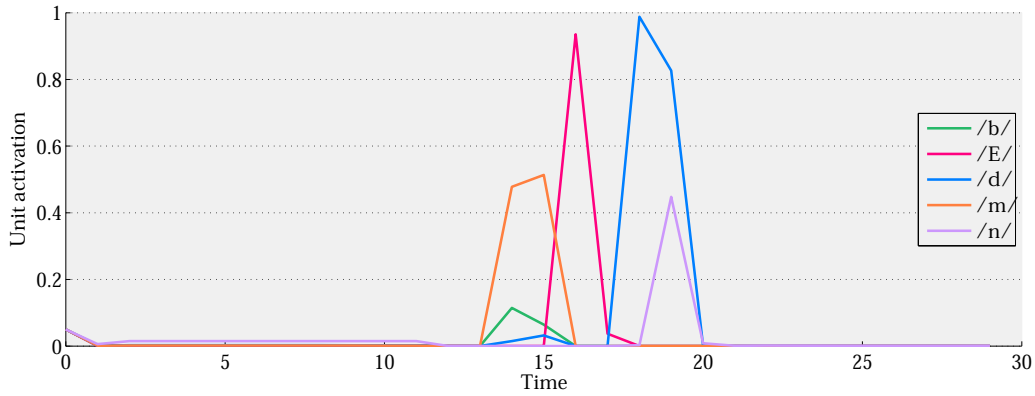


(c)

Figure 6.9: Panels (a) and (b) show the activation values taken over time by the first ten units in layer *H*3 under testing condition th_L and th_F, respectively; the first ticks, where all units are forced by the inhibitory unit to stay silent, are not graphed. The activation profiles are rather similar, but differences, mainly in the increase/decrease rate, do exist, as shown by the activation difference graph in panel (c).



(a)



(b)

Figure 6.10: The activations over time of five phoneme units, which achieve a minimum activation value in at least one testing condition, are plotted here. Under the th_L condition, the pronunciation of input word BEAD is correct (panel (a)). On the other hand, pronunciation errors occur in testing condition th_F (panel (b)), especially at the first output position.

do not reach the current training threshold. In the implementation described so far these cases were not counted as errors, as the produced phonemic sequence was in fact correct, but they are errors if we consider that for these items the network was not able to develop an internal representation that was strong enough to reach the response threshold. Ideally, it would be desirable that a signal was given to the network to encourage it to strengthen such a code, so that the threshold could finally be reached. To this end, a version of the model employing the utility unit (see Section 6.4.2) was implemented, as this version allows for the internal code to be more easily instructed to get stronger when needed, with respect to versions adopting quality measures based on the net input to layer *H*3. In this version, error is injected to the utility unit whenever the current threshold is not reached by an example within *maxSettlingTime*; note that this is done in training phase 2 only. This was achieved by setting a target for the utility unit, too; this is usually a void target, but is changed to 1

when needed (that is, when the current threshold has not been reached). The intended effect is that units in layer $H2$ are driven to activate more strongly to increase the quality measure values (recall that in this case the quality measure corresponds to the average activation level of layer $H2$), so that eventually no items are left for which the current threshold is not reached. Unfortunately, this version also seems to suffer from previous problems: lack of frequencies effects; low variance in the output times; difficult tuning of threshold values and training durations.

6.6 Conclusion: open questions and future directions

In this chapter, we have described different implementations for a proposed model of production (i.e., the mapping from the semantics of a word to its phonology), as a first step toward building a complete model for single word reading. The general framework is based on the assumption that a single, all-purpose internal code is computed, based on different sources of incoming information (orthographic, phonological, and semantic), which can then be used to drive different output systems, depending on the task at hand. For instance, for both production and reading aloud, the output system is the articulatory system that produces a sequence of phonemes constituting the spoken word. In the model, a cognitive part takes care of generating the code for the input word, so that this code gradually emerges from noise to become more and more strong and clear as more information is processed. Such code is processed, in a cascaded manner, by the articulatory part of the model, so that when the code is strong enough, a response can be safely initiated. Depending on how quickly the code for a word grows clear enough (i.e., how quickly the quality of the code crosses the response threshold), each input word will be pronounced with a different latency. The latencies produced by the model can then be compared to latencies reported in the literature for human readers in order to assess whether the relevant effects classically found in reading (e.g. frequency, frequency-by-irregularity interaction, consistency, etc.) are also reproduced by the model.

We have reported here some preliminary investigations that we carried out in order to assess the viability of this approach. By analyzing the results obtained in simulations based on the implementations detailed above, we could identify two major problems, namely a general lack of stability in the codes learned by the network, and the lack of significant frequency effects. Lack of stability was revealed when studying the behavior of the network at test time, when the response threshold was increased with respect to the value used during training: although it would be expected that no, or only mild, degradation of performance was experienced in this case, the opposite pattern was observed instead. The absence of frequency effects in response times (i.e., more frequent items are expected to be pronounced with faster latencies than less frequent ones) was consistently found in all versions of the model, although there were indications that a frequency effect on errors (i.e., accuracy is higher for high-frequency vs. low-frequency words) might emerge when testing the network in speeded mode, that is, by decreasing the response threshold and therefore causing an earlier response

than when in at-leisure condition.

Based on these observations, we can list some open questions about this model, and start addressing a few points that we think should be more deeply examined in future work:

- What strategies can be devised to force a more attractor-like dynamics in the model, so that learned codes are made more stable? Which are the key parameters affecting this aspect?

- Is the lack of significant frequency effects in the production task a major sign that this model is fundamentally flawed?

- Which training regimen is more appropriate and likely to produce consistent results: the fixed-threshold approach, or the adaptive one?

We have already mentioned that the lack of stability might be in fact a sign of overfitting. We attempted to reduce the problem by decreasing the number of hidden units in each layer, but this did not impact significantly on the observed pattern of results; in some experiments, we also introduced some degree of weight decay (from 0.00001 to 0.0001), but this either prevented learning or had no effect. The problem with code stability might also be a consequence of ill-defined quality measures. A quality measure that is strictly non-monotonic might perform better, for example. Another approach could involve coupling an absolute quality measure *with* a stability measure: that is, the network could be instructed to initiate a response when a value measuring the quality of the code crosses an (absolute) threshold, *and* this value has not been changing appreciably during the last ticks (i.e. a threshold on the quality measure variation over time is imposed, too). On the one hand, this correction might possibly improve stability, but it would also introduce additional assumptions on the response initiation mechanism, making it more complex and artificial.

We have also advanced that the stability problem might actually be a consequence of overtraining. Additional investigations on this point are required: here, we can hypothesize that less extensive training regimens (both in the fixed threshold and in the adaptive threshold case) could make the network more flexible and robust to variations in the response threshold. Moreover, overtraining could have contributed to mask frequency effects, as the network could possibly have been exposed to low-frequency items for a large enough number of times, to the point that all the dataset is perfectly mastered independently of differences in frequencies. It might also be argued that a different compression of frequency values than the one used here (see Eq. 6.3) might be useful in potentiating the effect of frequency in training the model (see Plaut et al., 1996, for a similar point). Finally, it is important to remark that words here were not represented by their orthographic form, but rather by an arbitrary semantic representation: in the case of the orthographic representations, the word-level frequency may be reinforced by frequency at sublexical levels (such as graphemes), whereas in our case such additional source of frequency was not available. For this reason, the implementation of the orthographic input section of the general framework (Fig. 6.1), besides leading to a more com-

plete reading model, could be relevant for a more grounded assessment of the potentialities of the model.

We can think of at least three different implementation approaches for the orthographic input section of the model: (1) the letters/graphemes in the input word might be all turned on in a parallel fashion, from the first tick of computation (for instance by adopting a slot-based representation); (2) each letter/grapheme might be activated separately, on subsequent ticks; (3) a fixed portion of the letter string, falling into an attentional window, might be presented to the network in parallel, with the window being shifted when a new portion of the string needs to be scrutinized. The influence of attentional mechanisms might also be implemented in the form of a "wave of activation" that moves from the left to the right of the string: the activation level of the letter currently being attended to would be maximum, with nearby letters being activated to a lesser degree. Sequential representations have the advantage to allow for an easy handling of string letters of arbitrary length, including multi-syllabic ones; in this latter case, however, a mechanism for stress assignment should also be implemented. Orthography and semantics would thus coexists in the extended model; to integrate them, one could think of training the network on both representations simultaneously, or to introduce orthography only in a second training stage, to simulate the delay of reading acquisition with respect to word production experienced by children. The model would also be enriched by the use of more informative semantic representations (for instance, based on specific semantic features).

As for the last point cited above, adaptive approaches generally tend to look more elegant and natural than those based on fixed, arbitrary parameters. In our implementation of adaptive training, the model has to learn by trial-and-error to set a response threshold so that it can be as fast as possible without compromising accuracy; in a sense, this regimen promotes an attempt at optimal resource allocation in response to a speed-accuracy tradeoff. Training on a fixed threshold might not impose enough pressure on fast computation, if a large threshold value is chosen; or, if a small value is chosen, and training time is limited, sufficient accuracy might never be achieved. We have seen how extensive training made it possible to have the network learn the whole dataset with virtually any response threshold: in order to simulate the effects of interest, however, it is necessary to limit the available computational resources – in other words, to make the task hard for the model. This can be done, as already mentioned, by limiting training duration and/or the number of hidden units. Empirically, one could find a reasonable size for hidden layers by training networks with successively less hidden units to produce immediate pronunciations within a pre-specified number of training runs: if the network succeeds, then the task is still too easy for it; otherwise, the network can be considered to be reasonably sized to correctly simulate reading/production tasks.

A third way might be considered, as well. It could be advanced that the ideal dynamic in the network should look like the one depicted in Figure 6.11: that is, internal codes for all items reach the same, maximum quality value in the same number of computation ticks. However, different codes are characterized by different growing rates, so that the response threshold is crossed at different points in time by different items. This does not happen in

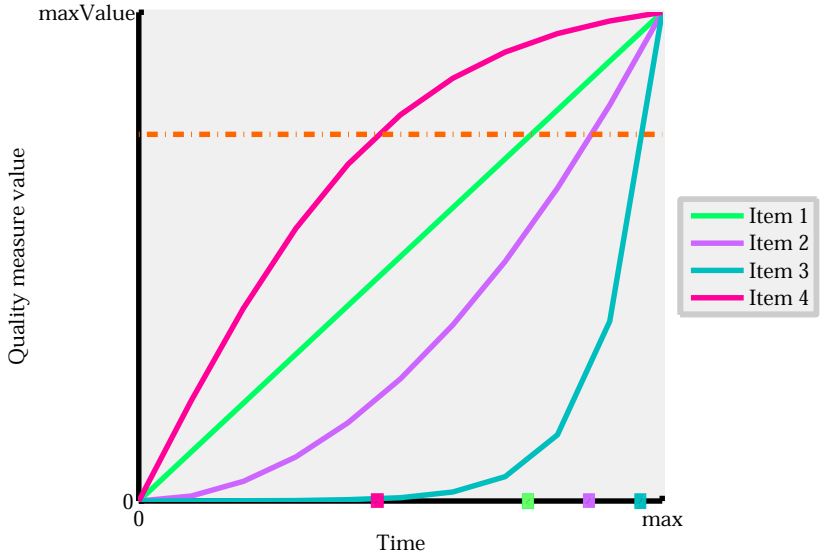


Figure 6.11: Hypothesized ideal course of processing for internal codes in the model. Values assumed by the quality measure over time are shown for 4 hypothetical items: values grow from 0 to *maxValue*, which is reached at the last time step; each item is characterized by a different growing rate. For this reason, any given response threshold (as the one represented here by a dotted horizontal line) will be crossed at different points in time by each item, thus resulting in different naming latencies (small squares on the x axis).

the current versions of our model, because each item reaches a different maximum value in the quality measure. To enforce such behavior, one could devise a training regimen where the network has to learn to correctly pronounce all items at *maxTime* and, at the same time, develop internal codes so that their quality measure reaches a maximum value at the end of a set interval. In this approach, the response threshold would come into play only at test time: in other words, training would occur without any time pressure, which would be introduced only when testing the network by setting the response threshold to different values, corresponding to speeded and at-leisure conditions.

We have therefore outlined some possible additional issues and future developments that we believe might be addressed to fully assess the potentialities and limits of the model we have started to study in our work. Even though, at the present state, the model was found to be affected by some weaknesses, we believe that different implementational choices (like the ones suggested above) might help solve them. Additional investigations are advisable as this model represents, in our opinion, an interesting attempt to offer a unified account of lexical tasks (e.g. reading, production, spelling, repetition, etc.) based on a shared, all-purpose code and on the assumption of cascaded processing between a cognitive core and the respective executory systems, thus tracing a new, hopefully fruitful branch in the ever-growing tree of computational models of reading.

Part III
Simulation of emotional interaction

Chapter 7

Emotions: the human perspective, and the machine one

"The question is not whether intelligent machines can have emotions, but whether machines can be intelligent without any emotions."
— From "The Society of Mind", by Marvin Minsky, 1927–

7.1 Introduction

We usually tend to identify human intelligence with our ability to reason, make deductions, perform calculations, operate complex machinery, speak, etc. – and we would call intelligent any artificial agent that exhibited similar abilities. That is, there is a general bias toward reducing human intelligence to its more rational, logical side. In fact, rationality is what distinguishes us from other animals. However, that does not exhaust, by any means, human intelligence.

Emotions may be psychological constructs hard to define in a formal way, but are nonetheless relevant components of human cognition. Far from being just annoying accidents that blur our rational thinking (although they do act like that, oftentimes), emotions coexist with rational cognition in ensuring our survival, helping us reach our goals, forming judgments on events and objects, learning, and remembering what we learned (Damasio, 1994). As our understanding of emotions and of their influence on our intellectual life has grown over years, the importance of the so called *emotional intelligence*[1] (Matthews et al., 2003) for a fully accomplished life has been increasingly acknowledged.

However, traditionally, both artificial intelligence and cognitive science have been focused on the rational side of cognition, mostly disregarding emotional aspects. The notions of emotions and feelings generally appeared to be too distant from, and hardly compatible with, those of computations and machines. It has been only recently that the interest of both fields has been directed to the influences that emotion exerts over cognition; in cognitive science,

[1] Emotional intelligence *"refers to the competence to identify and express emotions, understand emotions, assimilate emotions in thought, and regulate both positive and negative emotions in the self and in others"* (Matthews et al., 2003, p. 3).

models of emotional processes have been proposed (see for instance Boden, 2008), whereas in artificial intelligence a new research field came into being, *Affective Computing* (Picard, 1997), which aims at endowing computational devices with emotional competences. If we consider emotions to be just another set of neural events and mechanisms, as those involved in other cognitive domains, then there is no intrinsic prevention from treating them from a computational point of view – actually, considering the impact of emotions in the mental life of humans, and the explanatory power of computational methods, this looks like a very advisable approach.

In the last part of this Thesis, we turn to consider the topic of computational models of emotions. In the previous two parts, we have taken the cognitive science perspective, in using tools provided by computer science for advancing our understanding of human cognitive processes (specifically, single word reading/production). Here, we switch to an artificial intelligence standpoint, to illustrate how typically human abilities – in this case, emotional competences – can be incorporated into artificial agents in order to improve human-machine interaction, and to bring artificial intelligence a step closer to that human intelligence it takes as its inspiration. In this chapter, we will provide an overview of the role of emotions both in human cognition, as seen from a psychological, neuroscientific, and computational point of view, and in artificial agents, drawing a few examples from the now wealthy literature on Affective Computing. Finally, the next chapter reports our work on such topics: a model for human-robot, or inter-agent, emotional interaction is described, which is based on the mutual exchange of emotional states, and includes features inspired by the concepts of personality, attitude, and experience-mediated adaptation to the interaction partner.

7.2 Emotions, in humans

7.2.1 Theories of emotion

As clear and immediate the term *emotion* might appear, it may be surprising to realize that a widely shared definition of what an emotion is, can hardly be found. We have to resort to rather general descriptions like the one given in (Niedenthal, 2003): *"Emotions are sets of processes that involve different components including subjective feelings, but also expressive motor action, cognitive appraisals, physiological arousal, and tendencies to take particular actions"* (see also Scherer, 2005). The picture is made even more complex by the existence of other related terms that may be easily confounded with emotion, such as feeling – which is usually employed to denote only a component of emotion, namely its subjective experience – or mood – which is generally taken to be milder in intensity and more persistent in time than an emotion, and less event-driven. Difficulties in providing a simple, conclusive definition of emotion reflect the complexity of the subject, and the relative obscurity that still surrounds it.

Several theories of emotion (Frijda, 1986; Lazarus, 1991; Gray, 2001) have been proposed that present a different notion of what an emotion is, why it arises, and which effects it determines. Classically, "centralist" theories have been opposed to "peripheral" ones: in a way, the

former emphasize the role of cognition while the latter stress body reactions (Picard, 1997). The James-Lange theory (James, 1884) proposes that emotions are induced by physiological modifications, such as a change in the heart beat rate, triggered by a meaningful event; more precisely, it is advanced that *"the bodily changes follow directly the perception of the exciting fact, and that our feeling of the same changes as they occur is the emotion"* (James, 1884, pp. 189–190). Thus, according to this theory emotions are generated at the periphery of the nervous system, rather than at its core, from cognition. The opposite position is taken by the Cannon-Bard theory (Cannon, 1927), according to which emotions are generated centrally, by the brain, and the observed physiological changes are a consequence, rather than the cause, whose removal would not prevent the emotion from being experienced. A third position, somewhat mediating between the previous ones, is embodied by the Schachter-Singer theory (Schachter and Singer, 1962): while the feedback about physiological events following the perception of a meaningful stimulus determines a generic state of (more or less intense) arousal, it is the process of cognitive evaluation of the triggering stimulus that appends a specific label to such arousal, so that it is possible to determine the type of emotion being experienced (e.g. anger vs fear).

What is common to these theories is the idea that an emotion is, at the same time, a physical and a cognitive event; the relative importance of these two components changes across theories, and for single instances of emotional experiences. For example, in (Damasio, 1994) it is proposed that some emotions (labelled as primary) are characterized by an immediate physiological response (such as startling in response to a potentially dangerous stimulus), with cognitive evaluation occurring only later, whereas others (secondary emotions) depend on the contribution of cognition to be fully experienced, and can be triggered by internal events too (like thoughts and memories). Such a division is suggestive of the innate vs nurtured polarity (with primary emotions assumed to be innate, and secondary emotions emerging in later developmental stages), and of the existence of two different "emotional processing routes" (see Section 7.2.2).

The relevance of cognitive processing in emotion is stressed in appraisal theories (Scherer et al., 2001; Lazarus, 1991; Ortony et al., 1988): an emotion is experienced only after cognitive appraisal checks have been carried out on the relevance of the triggering stimulus or event with respect to the person's goals, beliefs, relations with the environment. For instance, in (Scherer, 1984a,b) the appraisal process is made up of a series of *stimulus evaluation checks*, which interest, respectively, the dimensions of novelty, intrinsic pleasantness, goal/need significance, coping potential, and norm/self compatibility standards. The result of these cognitive operations is the "selection" of the appropriate emotion: therefore, according to this view emotion *follows* cognitive appraisal. Different appraisal theories diverge in the specific cognitive checks that they suggest. For example, the OCC model (from the initials of the authors – Ortony et al., 1988) details the cognitive appraisal process by providing possible rules whereby 22 different emotions could be elicited, as valenced reactions to external events and objects: emotions are generated by evaluating the impact of external events on one's goals,

the adherence of one's actions to some standards, and one's attitude toward some considered object.

Evolutionary theories, on the other hand, draw attention to the adaptive function of emotions in supporting survival and evolutionary success (Gray, 2001). Fear determines acts of withdrawal from a possible source of danger and prepares the body for defensive action (either for fighting or for fleeing); in general, emotions act as motivators that can either reinforce or discourage a certain behavior. Darwin (1872) was the first to point out the role of emotional expressions (such as facial expressions, or vocalizations) in effectively communicating a set of essential, relevant messages within a social group, such as a disgusted expression for signalling a non-eatable substance. The evolutionary standpoint generally implies that some emotions, or at least their expressions, are innate; there is actually some evidence supporting this position, for example the fact that blind-born children appear to produce facial expressions similar to those displayed by non-blind individuals (Eibl-Eibesfeldt, 1973).

In the evolutionary perspective – but not limitedly to that – it is assumed that there exists a finite number of discrete emotions. Ekman (1984, 1992b,a) proposed a set of six basic, or universal, emotions (happiness, sadness, surprise, anger, fear, disgust), based on the observation that facial expressions used to convey them appear to be similar across cultures. They are also basic in the sense that they appear to be easily differentiable (based on their cognitive antecedents, and associated patterns of physiological change), and may be conceived as the building blocks of other, more complex emotional episodes. Other emotion theorists, however, support the view that emotions are defined over a continuous, multidimensional space (e.g. Schlosberg, 1954), where each dimension codes for a particular feature of an emotion, and each point in this space is a specific, different emotion. Typical examples of emotional dimensions are the degree of arousal (mild vs intense emotion) and valence (positive vs negative emotion) (e.g. Russell, 1980).

Any attempt at adjudicating between different theories of emotion is complicated by the intrinsic difficulty in observing and analyzing the emotions themselves: in a typical behavioral setting, the subject would be asked to provide a linguistic description – a label – of a current or past emotional experience. However, emotional labels used in everyday language are not standardized, come in a large variety, often with subtle differences between similar terms, and subjects usually differ in the range of their "emotional vocabulary". Alternatively, emotions can be deduced by observing their external expressions: these include body language (Argyle, 1975) and speech prosody, and changes in physiological parameters such as the heart beat rate, skin conductance, respiratory rate, and blood pressure. While both body language (gestures, posture, eye movements, and facial expressions) and speech intonation do not need invasive approaches to be analyzed, and are the main channels for emotions in real life situations, the study of emotion through the collection of physiological data is necessarily limited to laboratory settings, as it requires the application of sensors to the subject's body[2]. However, physiological parameters "do not lie", or at least are more difficult to ma-

[2]The introduction of wearable devices, however, has in part mitigated the discomfort of this measurement

nipulate than bodily expressions: in social situations, for instance, it is not uncommon to fake, or to suppress, facial expressions. Facial expressions are in fact one of the most effective, and most widely studied, channels for emotion communication. For studying them in a rigorous way, Ekman and Friesen (1978) proposed the *Facial Action Coding System* (FACS), which describes a set of basic facial actions, called Action Units (AUs), that can be seen as the elementary components of every facial expression. FACS can therefore be used as a way to formally describe an observed expression in terms of its composing AUs, their duration, intensity, asymmetry, and so on. However, it does not explicitly link groups of AUs to specific emotions, although some correspondences can be easily derived (for instance, AU12, corresponding to the lifting upwards of the corners of the mouth, can be associated with the expression of happiness).

The psychology of emotion is, as we have seen so far, a lively and wide-ranging area of research, where much still needs to be clarified. Additional insights on the mechanisms underlying emotion recognition, subjective experience, and expressions, can be derived by neuroscientific studies.

7.2.2 Neuroscience of emotion

What we know about the neural basis of emotion has mostly been derived by lesion studies performed on animals, and observations on brain-damaged patients. More recently, neuroimaging studies aimed at locating the neural circuits involved in emotional processing have started contributing to this body of knowledge (Davidson and Irwin, 1999). These studies collectively point to a set of cortical and subcortical structures, playing an important role in emotion-related operations (Bear et al., 2001; Gray, 2001; Umiltà, 1999). Originally, Papez (1937) proposed that the hypothalamus, the anterior nuclei of the thalamus, the cingulum, the parahippocampal gyrus, and the hippocampus formed a circuit implementing the "emotion system". As this set of areas largely overlaps with those structures that were named the "limbic system" by Broca (1878) – that is, the cingulate gyrus and the medial temporal lobe – this locution has in time become synonymical with that of emotional system, although there were no implications in this sense in the original definition by Broca. Each anatomical structure in the circuit of Papez participates in some aspect of emotional processing: the hypothalamus governs autonomic responses to an emotionally significant stimulus, whose evaluation is supposedly carried out by the thalamic nuclei; the hippocampus might be involved in contextual conditioning[3] and emotion-connoted memories, and the cingulum has been linked to emotion experience, and attention direction processes (Morrone-Strupinsky and Lane, 2003).

approach.

[3] In the classical conditioning paradigm (Pavlov, 1927), a normally neutral stimulus, such as an auditory tone, is presented every time a meaningful stimulus, such as food, or a shock, is delivered to the experimental subject. With repeated presentations the conditioned stimulus (CS – the tone) becomes associated with the unconditioned stimulus (US – the shock) so that the response originally triggered by the US is now transferred to the CS (that is, the subject starts to show a fearful behavior upon presentation of the tone). In contextual conditioning, the conditioned response is transferred also to the context (e.g. the laboratory room) where the CS and US were initially paired.

Evidence from later studies has enlarged the emotion system as originally described by Papez to two other anatomical regions: the amygdala and the prefrontal cortex.

The role of the amygdala, an almond-shaped structure composed of several nuclei lying deep in the medial temporal lobe, in emotional response and conditioning, especially for fear, has been extensively investigated in both animals and man. Kluver and Bucy (1937) performed a bilateral temporal lobectomy (which included the amygdala) on monkeys, following which severe behavioral modifications were observed: the monkeys appeared not to be able to recognize objects through vision, but had to examine them with their mouth, were affected by hypersexuality, and showed a strong reduction in fear and rage responses. A series of studies on rats by LeDoux (1996, 2000) revealed the central role of the amygdala in the analysis of potentially dangerous stimuli, in generating a fear reaction in response to them, and in coding, and possibly storing, associative emotional memories related to the conditioning episodes. These functions are supported by extensive connectivity from cortical and thalamic sensory regions (stimulus analysis), and to the hypothalamus, the brain stem (generation of appropriate autonomic and behavioral responses, respectively), and the cortex (subjective experience of emotion). The amygdala has also connections back to cortical sensory areas, both direct and indirect, and is thus able to influence sensory processing, possibly by facilitating the analysis of the salient stimulus.

The experiments on fear conditioning in rats also lead LeDoux to develop a theory of how potentially dangerous stimuli are processed (LeDoux, 1996) based on the existence of two anatomical pathways from sensory input to emotional response: the “low road” delivers quick, raw information coming from the thalamus directly to the amygdala, so that an immediate response (for instance, an avoidance behavior) can be initiated if necessary; the “high road” passes through the cortex, before reaching the amygdala, and, although slower, is able to convey more detailed information on the perceived stimulus. The two mechanisms work simultaneously, with only the high road being responsible for the conscious experience of the emotionally salient stimulus.

It has also been advanced that the amygdala is not only involved in fear conditioning, but might have a more general role in reinforcement processes, providing an assessment of the reinforcement value (either positive or negative) of an event and forming associations between event and reinforcement (see e.g. Rolls, 1999). The amygdala receives connections from the dopamine nuclei of the ventral tegmental area and the substantia nigra, which supports the view that reinforcement information may be processed here; dopamine is in fact believed to be the neurotransmitter responsible for reinforcement-based modulation in the brain (although it is also involved in action initiation). In fact, emotions may be interpreted as mechanisms activated by reinforcers for implementing a motivated behavior – that is, as drives toward achieving rewards and avoiding punishments. Under this view, the adaptive function of emotions appears clear.

Additional evidence of the role of the amygdala in emotion processing comes from studies on human subjects (Adolphs, 1999). Patients with focal lesions of the amygdala have been

reported to show difficulties in recognizing emotional expressions in faces, and emotional intonation in speech (Adolphs et al., 1994; Scott et al., 1997); in both cases, negative emotions, especially fear, seem to be the most impaired. Damage to the amygdala was also shown to prevent the acquisition of fear conditioning (Bechara et al., 1995) and to have potential effects on declarative memory too: although in normal subjects a memory enhancement effect is observed for highly emotional material (such as pictures with a disturbing content), this is not the case for amygdala-lesioned subjects (Adolphs et al., 1997). Neuroimaging experiments have contributed additional evidence of the role of the amygdala in emotion recognition (even if the stimulus is not consciously perceived because a masking paradigm is being used) (Whalen et al., 1998), confirming a greater involvement of this anatomical structure in the processing of fearful expressions. Evidence consistent with the view that the amygdala modulates memory for emotionally salient events has also been reported (Canli et al., 2000), showing that activation of the amygdala was correlated with the intensity of the emotional experience reported by the subject, and that the expected enhanced memory effect was predicted by significant activation of this region.

Another key role in emotion-related processes is played by the prefrontal cortex, as it was demonstrated by the classic case of Phineas Gage (Harlow, 1848, 1868; Damasio et al., 1994). Gage suffered an extensive brain lesion, interesting the prefrontal cortex bilaterally, when a tamping iron he was handling for his work as a construction foreman was suddenly propelled through his head. Amazingly, he survived the accident, but he showed dramatic modifications in his formerly controlled behavior: he became unreliable, irreverent, incapable of effectively plan his life, even though his intelligence, speech faculty, and memory were all preserved. It has been concluded that Gage's deficit was specific for emotional behavior control, determining the observed impairment in social competence and effective decision making; this case underlies the importance of the influence of emotion in making good decisions, showing that the absence of emotions impairs, rather than enhancing, rationality. The prefrontal cortex, thus, appears to be a planning and control center that coordinates emotional behavior, by possibly modulating the activity of the amygdala. The cortical contribution (mainly involving the prefrontal cortex and the anterior cingulate gyrus) have also been pointed to as being responsible for the conscious experience of emotion, perhaps as a result of information from different sources (perception of a stimulus, memory about previous experiences, and emotional arousal) being combined in working memory (LeDoux, 1996).

To sum up, the cortical components of the emotion system (the cingulum and the prefrontal cortex) are considered to be involved in the subjective experience of emotion and control of expressive behavior, with subcortical components (especially the amygdala, at least for the emotion of fear) responsible for carrying out rapid evaluation of the emotional content of the stimulus, and generating the appropriate behavioral and autonomic responses. The existence of bidirectional connections between the two macro-components of this system suggests that mutual influences are in place, with automatic responses being, at the same time, controlled by, and able to modulate, conscious processes (perception, decision making,

subjective experience). Thus, we can say that the anatomical picture we have drawn here is potentially compatible with both the James-Lange theory (body reactions cause the emotional experience) and the Cannon-Bard theory (cortical processing affects autonomic response).

7.2.3 Computational modelling of emotion

Emotions have been object of computational modelling efforts, although less extensively than other mental processes. However, the importance of emotion in controlling intelligent behavior was acknowledged from the very origins of cognitive science and AI, as witnessed by Herbert Simon's work (Simon, 1967). According to his view, emotions can be interpreted as interruption mechanisms that direct the organism's attention to urgent needs; in other words, emotions are resource allocators, and allow the system to simultaneously handle multiple goals. If emotions can be interpreted as functional mechanisms, then they are apt to be modelled as computer programs like other cognitive faculties. In this section we will present some examples of computational models that have addressed emotion-related issues.

Before presenting these models, we should stress that, as emotion is not a unitary concept, building an emotion model involves taking into account multiple aspects of emotion and related constructs. Emotions have a large subjective component, at least as far as their conscious experience, and behavioral expression, are concerned. For instance, the same emotionally salient event can determine different reactions in different individuals. Concepts like those of *personality*, and *attitude*, are for this reason strictly related to emotional behavior. Personality (Matthews et al., 2003; Ryckman, 2003) can be defined as a set of relatively stable characteristics of an individual that significantly influence his or her behavior; a widely agreed model for personality description is based on five dimensions, or traits, called "the Big Five" (Costa and McCrae, 1992): openness, conscientiousness, extroversion, agreeableness, and neuroticism. For example, an extroverted person is more likely to frequently engage in social situations than an introverted one. Social behaviors are influenced also by attitudes (Eagly and Chaiken, 1993), which can be seen as dispositions, beliefs, and opinions of an individual on specific persons, situations, objects; attitudes are essentially value judgments (as judging something to be good or bad) on some entity in the social world. Emotional information contributes to forming an attitude toward a given object, and it constitutes its affective basis in terms of the emotions that are linked to that object as a result of past experiences. Attitudes are more subject to change than personalities are, and their changes are driven by acquired experience with the attitude object. As attitudes serve, among others, an utilitarian function (Katz, 1960), they direct behavior so that entities toward which a positive attitude exists are actively pursued, while the others are avoided; in other words, attitudes guide behavior so as to maximize the rewards that can be obtained from the (social) environment. Because of their relations with emotional behaviors, personality and attitude are psychological variables that naturally get addressed in computational models of emotion. For instance, in (Mischel and Shoda, 1995) the Cognitive-Affective Personality System (CAPS) is proposed in the form of a network where different personalities corresponds to different (fixed) connec-

tion weights existing between nodes that represent specific beliefs, goals, plans, and affects; when a situation, coded as a set of binary features, is presented to the CAPS, this determines a (personality-dependent) activation of the cognitive-affective units and, in turn, a potentially different behavior.

After this due remark, we can start our review of models of emotion by briefly mentioning some interesting cognitive theories that, although their formulation naturally lends itself to computational modelling, were not accompanied by an actual computer implementation. Bower (1981) proposed a theory of long-term memory that includes emotion influences, based on the idea of a semantic network. According to network theories, concepts are represented as nodes of a network and are linked so as to reflect associations experienced during one's existence; when a node is activated by presentation of the corresponding concept, activation spreads to linked nodes, thus retrieving learned associations. Bower adds emotion nodes to the picture, which get activated when remembering events that elicited those particular emotional states; these, in turn, can activate other nodes that were previously associated with the same emotion. An interesting phenomenon that can be explained by this theory is mood-dependent retrieval: that is, it is easier to recall some piece of information when we are in the same mood as when it was learned. In network theory, this happens due to the spreading of activation from the emotion node corresponding to the current mood to the concept to be retrieved, the link between them being provided by the previously learned association. A theory according to which people are seen as self-regulating feedback systems has been described in (Carver and Scherier, 1990, 1998; Carver, 2006): discrepancy reducing feedback loops guide behavior by instantiating actions for approaching the desired goals (discrepancy enlarging loops are also postulated, which aim at increasing distance from undesired outcomes). Another level of feedback systems keeps track of the rate of progress in achieving goals, and whenever this is different (larger or smaller) from a criterion value, an emotion is experienced: thus, if we are approaching our goal less quickly than expected, sadness or anxiety may be experienced, resulting in actions aimed at increasing efforts. A related theory was proposed by Oatley and Johnson-Laird (1987, 1996): according to their "communicative theory", emotions are control signals that have the function to redirect attention and prepare the body for possible action. An executive system and a monitoring system work together, with the former running plans to achieve goals, and the latter evaluating goal-related events. Whenever the probability of attaining the goal changes (either increasing or decreasing), an emotional signal (corresponding to one of five basic emotions, each having a different effect on processing) is generated by the monitoring system to instruct the executive system to take appropriate actions, such as persevering or changing goal.

Some computational models have addressed specific (negative) emotions, and emotional disorders, such as anxiety and depression. Ingram (1984) described a theory of depression inspired by the network theory of emotion (Bower, 1981, see above). In this network, beside units representing cognitions about past and current events, a special depression node exists: when the activation of this node, as determined by cumulative appraisals of negative life

events, exceeds a threshold, depression results. As the depression node has become associated, in time, with a set of cognitive units related to this affect (such as events in the past that caused a depressed feeling), its activation spreads to these units as well, causing the corresponding cognitions to be brought into conscience. These, in turn, feed back to the depression node, causing the depressed feeling to persist in time, until the process of activation decay is not able to interrupt such "cognitive loop". This proposal was further elaborated in a localist connectionist model (Siegle and Ingram, 1997) that aimed at explaining observed differences in the experience of depression in dependence of personality variables (e.g. ruminative vs distractive coping processes). The model hosts ten units representing perceptual characteristics of a stimulus, connected to a layer of ten other units representing its semantics; finally, two nodes representing the positive and negative valence of the stimulus, respectively, have bidirectional connections with the semantic nodes. This network was trained by backpropagation to learn a set of nine stimuli (equally subdivided among positive, negative, and neutral ones). Depression was simulated by additional sustained (i.e. lasting for many epochs) training on a single negative stimulus, on the assumption that the onset of depression depends on a major negative event that is constantly brought into conscience; when tested on an affective valence identification task, the "depressed" network was shown to identify negatively-valenced items faster than positively-valenced ones, similarly to experimental subjects. By changing learning parameters and conditions different personality variables were simulated (e.g. increasing the number of semantic-affective feedback iterations to simulate rumination) and their possible impact on the onset, maintenance, and recovery from depression was demonstrated.

Wells and Matthews (1994, 1996) have proposed the Self-Regulative Execution Function (S-REF) model of emotional disorder, which is organized in three interacting levels: a first level devoted to process external information, thoughts, and body changes in an automatic way; a second level of information processing, characterized by attention-demanding and voluntary operations; and a final level encoding self-beliefs. The S-REF can be seen as a module within the voluntary processing level, which performs self-regulation processes aimed at reducing the discrepancy between the desired goal and current state; this is done according to plans (encoded as self-beliefs) that drive the appraisal of external and internal events (thoughts), direct attention, select memories, modulate the activity of lower level nodes, etc. Emotion results from failure in reaching the goals posed by the plans. It is proposed that some processing strategies within the S-REF, such as rumination ("active worry"), can determine emotional disorders, as they subtract attentional resources from other activities not concerned with the dysfunctional beliefs and, therefore, reinforce these latter. In particular, the authors suggest that the implementation of a voluntary threat-monitoring strategy within the S-REF (whereby the selected plan directs attention toward specific kinds of information that are believed to represent a threat) can explain attentional bias in cognitive tasks such as the Emotional Stroop task; that is, attention is drawn to words whose meaning is related to the specific emotional disorder of the individual, resulting in degraded performance in naming the ink color of those words. Although the S-REF model was not directly implemented,

the hypothesis of attentional bias as deriving from threat-monitoring mechanisms was tested in a related modelling work (Matthews and Harvey, 1996); there, a connectionist network for the Emotional Stroop task was designed, where inputs were coded in terms of their color, semantic properties, and emotional content; two units were used to signal task demands (either color or word naming), and one unit represented the threat-monitoring strategy, so that when this was on, only emotional words had to be responded to. Simulations on this model reproduced a pattern of interference effects on the color naming task that provided some support to the attentional bias hypothesis set forth in (Wells and Matthews, 1994, 1996).

An emotional architecture similar to the one proposed in the S-REF model is described in (Wright et al., 1996), with low-level automatic processes (such as sensory analysis, reflexes, associative memory), management processes that must cope with limited resources, and meta-management processes to control them. Desires and goals are motivators that are processed by the management system to decide which plan should be executed, and which priorities are in place; but, as attentive resources are limited, not every motivator and task can be simultaneously processed. In particular, only *insistent* motivators are able to pass over an attentional filter and be considered by the management processes; the attentional filter is a dynamic threshold-based mechanism, whose threshold changes according to context and to the number of concurrently active motivators. The authors showed how such architecture could possibly explain the grieving state following the loss of a loved one. This is interpreted as a perturbant state, that is a partial loss of control of attention due to the constant presentations of thoughts about that person, and the need to massively reorganize information involving them; this prevents other goals to be achieved because of saturation of the attentive resources, which, in normal conditions, would be contrasted by the attentional filter. As perturbant states often characterize emotional states (for instance, great excitement can interfere with the ability to focus on work activities), emotions are here seen as affecting the scheduling of attentive resources in the cognitive agent, much in the spirit of Simon (1967). In (Wright and Sloman, 1997) a partial implementation of the proposed motive-processing architecture is presented: MINDER1 is an agent moving in a highly attention-demanding domain, as it must take care of "minibots" that constantly wander around and can run into troubles such as falling into ditches. For this reason, MINDER1 is often in need of managing multiple motives (that is, taking care of the needs of many minibots) while having limited resources, which, under some circumstances, can determine the emergence of perturbant states (we can interpret them as "anxiety" for having to take care of too many conflicting goals).

Although existent computational models of emotion are mainly symbolic, and conceived at a rather high level of abstraction, few anatomical-inspired models have also been proposed. Armony et al. (1997) developed a connectionist model of auditory fear conditioning in rats, organized in modules corresponding to the hypothesized anatomical structures involved (the thalamus, the amygdala, and the auditory cortex). Modules are connected by feed-forward connections that mirror the two roads postulated by LeDoux (1996). Both the conditioned stimulus (auditory tone) and the unconditioned stimulus (aversive signal) are presented to

the network, and conditioning is acquired by Hebbian learning; the winner-takes-all dynamics within each module result in units having a frequency-based receptive field centered on a preferred frequency that, after conditioning, can become biased toward the conditioned tone. Interestingly, lesioning the network at the level of the auditory cortex produced a counter-intuitive prediction (the expected generalization of the fear response to other tones than the conditioned one was not observed) that was confirmed by an experimental study; these findings have been taken to suggest that the "low road" may in fact be capable of finer stimulus analysis than it was previously assumed.

The relationship between cognition and emotion has been investigated in several modelling works. In (Thagard and Nerb, 2002), emotion and cognition have been integrated into a coherence theory of decision-making and judgment formation referred to as emotional coherence. The decision-making process is modelled as a constraint satisfaction network, where units represent concepts whose plausibility is proportional to their activations, and connections represent constraints – that is, a positive connection exists between two units if the related concepts are compatible, whereas a negative link connects two mutually exclusive units; activations spread as in classical connectionist networks. The process of constraint satisfaction involves the selection of units to be active so that the largest possible number of constraints is satisfied[4]. A computational model named HOTCO has been proposed (e.g. Thagard and Nerb, 2002) to explain "hot coherence" (or emotional coherence), that is cognitive coherence that involves also motivations and emotions: the constraint satisfaction network is augmented by endowing each unit with a valence variable that is updated based on other valence values flowing in from connected units, in a similar way as to how activations are computed. The valence judgment for a representation reflect the individual's attitude toward it. Valence values also participate in unit activation updates, so that the acceptance of a representation is made in part dependent on its desirability (as in wishful thinking); in this sense, emotion affects cognition. The result of computation in this model is an evaluation of the current situation, based on constraint satisfaction; emotion can be interpreted as a consequence of this evaluation, based on one's goals, as advanced in appraisal theories. Emotional coherence theory was applied to specific practical scenarios: in (Thagard, 2003) the HOTCO model was used to show how emotional coherence could explain the (largely unexpected) jury verdict about O.J. Simpson murder trial, while in (Sahdra and Thagard, 2003) self-deception was modelled; in (Nerb and Spada, 2001) a model similar to HOTCO, ITERA (Intuitive Thinking in Environmental Risk Appraisal), characterized by the presence of specific emotion units (for sadness and anger), was employed to predict distinct emotional responses elicited in the public in response to reported environmental problems, when different kinds of information were delivered by the media (for instance, if the environmental risk was reported to have been possibly prevented by adequate measures that, however, were not taken, then ITERA predicts that anger, rather than sadness, will be experienced).

Several models are in fact implementations of appraisal theories of emotions. EMA (EMo-

[4] A constraint is satisfied when both, or none, representations at the ends of a positive link are "accepted", that is, strongly activated.

tion and Adaptation – Gratch and Marsella, 2004; Marsella and Gratch, 2009) assumes appraisal to be a fast, parallel, and unique (rather than multi-process, as usually implied by appraisal theories, Scherer et al., 2001) operation that receives input from other (perceptual, cognitive) processes that perform inferences about the individual-environment relationship. As these processes are characterized by different processing speeds, re-appraisal can result in either fast emotional reactions, or slowly rising and changing responses. Patterns of appraisal outcome are linked to specific emotional states, which in turn engage coping strategies (i.e. threat avoidance); appraisal is also affected by the current mood that can bias the process toward some emotional states. The result of coping processes are control signals, either in the form of internal processes (for instance, abandoning an impossible goal) or external actions, which can update the current person-world relationship; beliefs, plans, intentions constituting such relationship are coded in propositional form. The model is shown to potentially account for the dynamics of emotion responses that can be observed in naturalistic settings. Several other appraisal-focused models have been developed (e.g. Marinier et al., 2009), which differ mainly in how they compute the intensity of an emotional response based on the appraisal over the considered cognitive dimensions. Some of them are more AI-oriented, in that these models aim mainly at improving the believability of artificial agents rather than faithfully simulating human data (and underlying processes). For instance, THESPIAN (Si et al., 2009) is employed in virtual characters populating interactive narratives: appraisal processes have also a social focus, as they can result in emotions related to other agents' goals. In FLAME (Fuzzy Logic Adaptive Model of Emotions – El-nasr et al., 2000) external events are evaluated based on the agent's goals, and fuzzy rules are used to compute a desirability value for the current situation and to derive from it an emotional state, its intensity, and ensuing behavior; various forms of learning (e.g. reinforcement learning) are also implemented to allow the agent to autonomously adapt to its user and environment[5]. Thus, these models can be either treated as cognitive models of emotions, or modules for "emotional" artificial systems: in the next section, we will shift our perspective to take on the latter interpretation, and review a selection of works in the Affective Computing field.

7.3 Emotions, in artificial agents

We have argued that human emotions are not annoying disturbances that impair our rationality; on the contrary, they serve a functional, adaptive purpose in guiding behavior and decision-making, in integration with purely rational cognition. Given this observation, it is not hard to see how artificial agents could benefit from being endowed with some emotion mechanisms. In particular, the ability of an emotion signal to quickly redirect attention to features of the world that require urgent action, and to support decision-making by guiding

[5]Both THESPIAN and FLAME share some features with our own modelling work described in Chapter 8, namely, the idea that the agent maintains an internal "model" of the interacting agent to be able to adapt to it, and the use of reinforcement learning in (El-nasr et al., 2000) to learn the utility of given behaviors with respect to one's goals.

the process of salience attribution when several variables have to be considered, looks like a very desirable skill to be transferred to our computational systems, which are, like human beings, often faced with problems characterized by an unlimited solution space while having only limited computational resources available. The role of emotion in consolidating and retrieving memories, in creating motivation, and in facilitating social interaction confirms the potential advantage of including emotional processes into computational devices.

Affective Computing (Picard, 1997) has exactly this aim: taking the beneficial effects of emotional processes and transferring them to computers[6]. As a result, affective agents are expected to be more adaptive, more autonomous, more skillful in interacting with users – in short: *more intelligent*, as their way of "thinking" would be closer to the human way, the intelligent being par excellence.

It should be remarked that, in the Affective Computing community, the focus is mainly on the functional role of emotion in getting better (under some respect: for instance, in flexibility, or in believability) artificial agents. Which specific emotional mechanisms are introduced, and how they are implemented, depends here much more on what is expected to be effective in the particular application setting, rather than on faithfully mimicking what is known, or assumed, about the human emotional system. Churchland and Sejnowski (1994, ch. 1) remarks that *"the computational solutions evolved by Nature may be quite unlike those that an intelligent human would invent, and they may well be neither optimal nor predictable from orthodox engineering assumptions"*. Thus, an intelligent artificial system does not necessarily need to mimic the human brain and its processes; it might well be possible that different strategies than the ones implemented in a natural system turn out to be more efficient for artificial systems. The perspective is therefore rather different from the one presented in the previous section (although some models of emotion actually lie at the boundary of the two approaches).

Research efforts in Affective Computing have translated in recent years into numerous affective systems than can be roughly categorized based on the (main) perspective they take on emotion: thus, we have systems that *recognize* emotions, systems that *express* emotions, generally for human-machine interaction purposes, and systems that *have* emotional processes that sustain cognition and behavior. In other words, we can distinguish between systems that treat emotions as their input, output, or internal processing modules. In an even more parsimonious categorization, a distinction can be made between systems that focus on human emotion, and those that focus on their own emotion. Following this classification, in the next sections we will describe representative work in each of these sub-topics[7].

[6] As it is often the case with technology, we cannot expect all the effects of introducing emotional components into artificial systems to be beneficial. Possible concerns could be raised, for example, about privacy – the user's emotion might be inferred even if he is not willing to overtly communicate it –, and the possibility that "uncontrolled" emotional systems are obtained, acting irrationally and therefore unreliably. Also, if a computer that can actually *feel* emotions is ever built, major ethical questions would have to be dealt with (as beautifully portrayed in the 2001 movie *Artificial Intelligence*). Designers of affective computers should be aware of, and take into account, these concerns so as to minimize the negative impact of affective technologies; however, it should be remarked that the current state of the art in this field is far from achieving such accuracy levels in the recognition and simulation of emotions in machines to make these worries immediate.

[7] Notice, however, that these classes of Affective Computing systems need not be mutually exclusive: for in-

7.3.1 Recognizing human emotion

Although we speak of "emotion recognition" (for reviews, see Pantic and Rothkrantz, 2003; Pantic et al., 2005; Jaimes and Sebe, 2007; Zeng et al., 2009; Calvo and D'Mello, 2010), we really mean "emotion inference". In fact, the affective states (including emotions, moods, attitudes) of a person are not directly observable, neither by computers nor by other humans. They must be inferred on the basis of external displays that, however, are not always reliable (for instance, facial expressions can be faked), and are often subtle and ambiguous. Emotion recognition is therefore a hard task even for people; it is so for machines to an ever greater extent (and in fact in order to achieve reasonable accuracy, a set of limitations on input images, such as no glasses or facial hair, a frontal pose, and ideal lighting conditions, is usually imposed), at least at the current state of the art, although significant improvements have been made in recent years (Zeng et al., 2009).

As mentioned above, information about the experienced emotional state is usually conveyed via multiple channels: the face is considered to be the most informative channel in this respect, and the one where the majority of research work on affective recognition has focused. Other channels that have been investigated in Affective Computing applications are speech prosody, body gestures and posture, and physiological parameters. All these channels, both in isolation and in combination, provide affective signals: to perform emotion recognition, a mapping must be found from these signals to the corresponding emotional states. In other words, emotion recognition is an instance of pattern classification (Duda et al., 2001), and in fact in most cases classical classification approaches (such as artificial neural networks, Bayesian classifiers, etc.) are employed in this domain as well.

Facial expressions

A wide variety of facial expression recognition methods (Pantic and Rothkrantz, 2000; Fasel and Luettin, 2003) can be found in the literature, differing with respect to

- whether they recognize a fixed number of basic emotions, or adopt a dimensional code for emotions;
- whether they recognize basic facial actions such as Action Units (Ekman and Friesen, 1978), which can then be mapped to affective expressions, or directly prototypical emotional expressions (for a review focused on the recognition of facial actions, see Donato et al., 1999);
- whether they process only static images, or videos;
- whether they perform manual or automatic feature[8] extraction;

stance, a system for human-machine interaction may both perform human emotion recognition and display an appropriate emotional state to the user, thus treating emotions as both inputs and outputs – this is also the assumption behind our interaction work (see Chapter 8).

[8] In this context, informative facial features can either be region-based, such as a whole eye, or on a smaller scale, such as the corners of the mouth (see for instance Jaimes and Sebe, 2007).

- whether feature extraction is performed holistically (i.e. considering the whole face) or locally;
- whether they are image-based or model-based approaches.

Many others distinctions are possible (e.g. adoption of 2D vs 3D features, real-time vs off-line processing, etc.), and for virtually any combination of the above possibilities a wide variety of algorithms have been proposed, which makes an exhaustive review of facial expression recognition methods, at the least here, impracticable. However, some representative examples can provide a flavor of the diversity of the approaches to facial expression detection that have populated the literature.

EMPATH (EMotion PATtern recognition using Holons – Cottrell and Metcalfe, 1991) is a system for recognizing facial expressions (and discriminating identities and gender) that uses neural networks for both feature extraction and classification. An autoencoder network (Cottrell et al., 1987, see Chapter 4, p. 114) was used to extract 40 holistic features from the input images, depicting 10 male and 10 female subjects performing eight facial expressions, approximately corresponding to groups of emotional labels defined on an arousal-pleasantness emotion space (Russell, 1980); these features, corresponding to the activation levels of the hidden units in the autoencoder, were called *holons*. Holon vectors were then fed to a separate feed-forward network with 20 hidden units and 8 output units (one for each emotion label): recognition accuracy on the training set was mixed (ranging from 5% hits on "boredom" to 100% on "astonishment"), and generally biased toward positive states. What is especially interesting here is the use of a compression technique for automatic feature extraction: this actually corresponds to performing PCA on the original input image, and results in representations that share similarities with the eigenfaces described in (Turk and Pentland, 1991), which can be seen as the most relevant variations occurring in face space with respect to a reference mean face. While in (Cottrell and Metcalfe, 1991) individual images are represented as projections onto the eigenfaces (holistic approach), in (Padgett and Cottrell, 1997) a local approach is introduced: image representations are now projections on eigenfeatures, which are the principal components of the covariance matrices computed over relevant sub-images (corresponding to the eyes and the mouth of the subject); the generalization results suggest that local feature extraction methods may be more powerful when it comes to expression recognition (as opposed to identity recognition, for example).

While EMPATH aimed at directly recognizing a set of emotional expressions, the system proposed by Essa and Pentland (1997) codes facial actions. It does so by resorting to estimation of optical flow coupled with a 3D model of facial structure that incorporates information on physical characteristics of the skin, muscles, and geometry of the face. The face image is automatically warped to the model mesh, and the motion of the identified feature points is computed between subsequent frames. Each expression of interest can then be paired to a 2D motion energy template to be used for classification purposes; by employing this method the authors report recognition rates on five expressions of about 98%. Moreover, the method is quite robust to variations in lighting conditions, and can handle face images with glasses

or facial hair. Another optical flow-based method is the one described in (Yacoob and Davis, 1996) to directly recognize expressions corresponding to the six basic emotions according to Ekman; here, a manual feature extraction procedure is performed to place bounding boxes around facial features (for instance, an eye). Then, the motion of whole rectangles is tracked between consecutive frames by taking the centroid of the high gradient points within a box as the reference point; a rule-based system maps the detected motions into emotional expressions, taking also the dynamics of the expression (that is, its onset-apex-offset evolution) into account.

In (Lanitis et al., 1995, 1997) a model-based approach is used where a 2D flexible shape model and an appearance (gray-level) model are paired. After manually placing, on each image in the training dataset, 152 landmark points that follow the profile of the face and of its main features, a mean model is created, and the eigenvectors of the correlation matrix of the dataset are computed; each new face can be modelled as the particular deviation from the mean image corresponding to a given set of parameters. These are found by fitting the flexible shape model to the new image. The use of a flexible model makes this method robust to changes in pose and lighting, and to partial face occlusions, and allows it to be successfully employed for multiple tasks, such as identity recognition, pose recovery, and expression recognition; in this latter case, a mean image for each basic emotion was reconstructed, and each new image was assigned to the emotional expression (the basic emotions were considered) whose parameter vector was minimally distant from the parameter vector computed for the test image. The process yielded 74% recognition accuracy.

Another system for facial action recognition is the one proposed by Tian et al. (2001). Here, multi-state local models are defined (for instance, the model for an eye has two states, open and closed) for both permanent and transient (such as furrows) facial features; the detection of the face in the first frame of a video sequence, and the approximate placing of the geometric models, are performed automatically, and followed by manual adjustment. For subsequent frames, each facial feature is automatically tracked based on the model shape, and (in the case of the lips) color information. Parameters corresponding to the tracked features are then fed to two ANNs that separately classify upper face and lower face AUs, either occurring in combination or alone; recognition rates around 95% were reported by the authors. The system also showed good robustness to head motion occurring in the frame sequence.

A more recent approach to facial expression recognition, which is able to operate in real time, is the one described in (Anderson and McOwan, 2006); in order to ensure real-time processing, constraints on the input images are introduced (namely, frontal views of the subject, and fixed scale). An algorithm for automatic face detection, based on the spatial relations between face regions (an appearance-base template is used to this end), is paired with an optical flow technique to compute face motion; average motion information computed over relevant face regions is input to six classifiers (Support Vector Machines using radial basis function kernels) that perform recognition of the basic emotions, with a recognition accuracy of about 80%. Notice that real-time processing is a very important factor for these facial expression

recognition systems to be effectively used in human-machine interaction scenarios.

Finally, recent work has also started to investigate different subjective states than basic emotion: for example, Ashraf et al. (2009) proposed a system for automatic detection of pain based on facial expressions of patients. An Active Appearance Model that combines both shape and appearance information is used to track expressions, starting from manually positioned templates on few key frames and then by automatic fitting. The model parameters are then classified by a Support Vector Machine, achieving in the best case a 82% hit rate (and 30% false alarm rate). This example highlights the potential utility of Affective Computing systems in assistive contexts.

Speech

Although less numerous than facial expression recognizers, systems for detecting emotional cues in speech have been developed too (Pantic and Rothkrantz, 2003; Ververidis and Kotropoulos, 2006). The most common approach consists in extracting acoustic features from the speech signal that are then classified into one of a number of basic emotions, or other subjective states. Examples of widely used features are pitch (characterized by the frequency of vibration of the vocal cords), intensity (or energy), and speech rate (measured in terms of voiced parts of speech per time interval). For instance, speech spoken by a happy person is usually associated with increased pitch, intensity, and rate with respect to a neutral state, whereas a sad person would display decreased values for these features (Pantic and Rothkrantz, 2003). However, recognition of emotional states from speech is not very accurate even when performed by human subjects (ranging between 55% to 70% when dealing with six possible options, Pantic and Rothkrantz, 2003), and thus accuracy for artificial systems cannot be expected to be perfect, either. Nevertheless, accurate feature selection can allow for good recognition rates to be achieved, as demonstrated in (Oudeyer, 2003) by using a genetic algorithm for finding an optimal set of acoustic features: the performance of different classifiers was compared and shown to range between 75% and 95% correct classifications. Notice, however, that emotion recognition was here performed in a speaker-dependent fashion, so the difficulties inherent to inter-subject variability were canceled out. Indeed, studies that did perform across-speaker classification reported lower recognition rates. For instance, in (Huber et al., 2000) anger recognition in a call center context was performed using a combination of annotated prosodic and part-of-speech features, and employing a multilayer perceptron as classifier: the achieved accuracy rate on novel speakers was just 66%, confirming the difficulty of the task of emotional speech recognition when generalization to other speakers' utterances is attempted.

A few examples can be useful for illustrating possible approaches to emotional speech recognition. One of the earliest works in this field was that by Dellaert et al. (1996), where statistics on the smoothed pitch contour (that is, the signal profile) were fed to different classifiers that were trained to recognize four emotional states (happiness, sadness, anger, and fear). The optimal error rate was of 20.5%, corresponding to a classifier based on majority

voting of specialists for subspaces of features. Amir and Ron (1998) proposed a real-time method based on the computation of relevant speech features (pitch average and variance, tremor, intensity, etc., all normalized by the respective baseline values corresponding to neutral speech) over sliding windows of few seconds, so that recognition could occur continuously (20 times per second). A reference feature vector for each of five emotions (happiness, sadness, fear, anger, and disgust) was computed, and the Mahalanobis distance between each reference vector and the currently tested vector used to build an "emotional index" that is the equivalent of a fuzzy membership value. This allows for a fragment of emotional speech to be associated with multiple emotions.

Works like the one by Litman and Forbes-Riley (2004) underline the potential utility of emotional speech recognition in enhancing the effectiveness of dialog tutoring systems: indeed, if the tutoring system is able to detect affective states such as frustration, or boredom, it can change its teaching strategy to better adapt to the student's needs. Here, both prosodic and lexical information is used to distinguish among classes of emotional speech (negative vs positive vs neutral, negative vs non-negative, and emotional vs non-emotional); interestingly, the introduction of an additional set of "contextual" features (about the identity and gender of the student, and the problem they are asked to solve) improves recognition accuracy, whereas prosodic features appear to be less relevant than lexical ones for emotion classification. The influence of lexical and paralinguistic (prosodic features, disfluencies in speech, laughters and cries, etc.) information on emotion recognition was also investigated in (Devillers and Vidrascu, 2006) using a corpus of dialogs from a medical emergency call center: the classification results appear to confirm the higher reliability of lexical information ($\approx$ 78% accuracy) with respect to paralinguistic features ($\approx$ 60%).

Finally, a very recent work (Sobol-Shikler and Robinson, 2010) has considered the problem of recognizing multiple co-occurring affective states. A large number (173) of vocal features is automatically extracted by the input spoken utterances, normalized in order to compensate for inter-speaker variability, and used to train binary classifiers, one for each pair of considered affective states (9 states were included). Each classifier also determines an appropriate subset of features that allows for satisfactory performance. A voting machine then processes the results of these classifications, by ranking each affective state based on the number of classifiers that have output that state; all co-occurring dominant states can thus be identified by setting a threshold on the resulting ranked lists. Although emotion recognition systems (in any modality) usually aim at detecting one affective state at a time, blended displays of emotion do occur in real life; methods like the one in (Sobol-Shikler and Robinson, 2010), or possibly fuzzy approaches (e.g. Austermann et al., 2005), might prove useful in recognizing such situations.

Other affective channels

Emotion recognition systems based on other channels than just face and speech have been proposed as well. For instance, Picard et al. (2001) proposed a system for the analysis of the

affective information carried by physiological signals. Four signals (elecromyogram from the masseter, blood volume pressure, skin conductance, and respiration rate) were recorded from one subject while she experienced eight emotional states in a predefined sequence; data collection was repeated every day for a month. Using a total of 40 features and Maximum A Posteriori classification, the system achieved 81% accuracy in discriminating among the eight considered states. Emotion recognition based on physiological data may be more intrusive and less comfortable than, for instance, facial expression analysis (although the introduction of wearable devices has in part alleviated these concerns), but it may also be more truthful as physiological signals are less prone to conscious control. In (De Silva et al., 2006), body gestures are collected from children playing a video game while wearing a motion capture system; after training, the employed Hidden Markov Model was able to discriminate among four affective states with an accuracy of 79%, operating in real-time. Gesture-based recognition systems could be used to obtain a more interactive game experience, where the difficulty level of the game can be changed to adapt to the user's state (frustrated vs bored, for instance). Some works (Kleinsmith et al., 2005; Kleinsmith and Bianchi-Berthouze, 2007) have focused on posture-based recognition of affective states: postures were collected via motion capture, and a network was trained on both form and motion features, achieving an average recognition rate of about 70% on four emotional states (Kleinsmith et al., 2005). Moreover, postural features were investigated to find the most salient ones for discriminating between intensity levels (e.g. high vs low) of each considered emotional dimension (specifically, valence, arousal, potency, and avoidance), thus providing the basis for a more precise recognition of affective states from postural data (Kleinsmith and Bianchi-Berthouze, 2007).

In some cases more than one affective channel is analyzed: these are referred to as multimodal affective recognizers (Pantic and Rothkrantz, 2003; Pantic et al., 2005; Jaimes and Sebe, 2007). Multimodality would actually be the final goal for an emotion recognition system, as affective cues from different sources can convey augmented information, and help disambiguate the particular emotion being expressed; after all, emotion recognition in humans *is* multimodal. However, currently few multimodal approaches have been proposed, if compared with unimodal methods. Most of these works consider audiovisual information: for example, Zeng et al. (2006) and Sebe et al. (2006) reported, respectively, $\approx$ 83% and 90% accuracy in recognizing 7 basic emotions and 4 cognitive states from facial and vocal features, using for data fusion, respectively, three component Hidden Markov Models, and a Bayesian network. Other modalities can be combined in an affective recognition system, as it is the case for the system described by Valstar et al. (2007). Here, the problem of discriminating posed from spontaneous smiles based on facial actions and head and shoulders movements is investigated. The head, and a set of reference points on the mouth, the eyes, and the shoulders, are automatically tracked along the frames of the input video sequence, and their motion is temporally segmented into onset, apex, and offset phases. Features from the three modalities are then combined following three different strategies: in early fusion, all features are simply concatenated to form a vector (actually, one vector per frame) that is then fed to the classifier;

in mid-level fusion, features are first turned into higher level attributes, and these enter the input vector for the classifier; finally, in late fusion, a different classifier is trained for each different high-level attribute, and their results combined to produce the final classification. The results of this work show that the discrimination between acted and spontaneous smiles is possible with high reliability (94%) if all three modalities are combined, especially with a late fusion strategy; also, the temporal structure of the affective signal is found to be particularly salient for this discrimination task, suggesting that the dynamics of emotional expressions should be taken into account to achieve finer classification capabilities.

Ideally, a system for affect recognition should be multimodal, robust (i.e. not sensitive to particular light conditions, occlusions, noise, etc.), person-independent, sensitive to the evolution in time of the affective expression, and context-sensitive, as many expressive signals can be disambiguated only by integrating knowledge about the context (for instance, to distinguish between a faked expression of anger from a friend, or a spontaneous one) (Pantic and Rothkrantz, 2003). However, this ideal set of features is still far from being achieved. As we have seen, multimodal approaches are still sporadic, and often do not perform a real integration of the different affective signals but just combine the results of their independent classification; good recognition accuracy can be assured only when strict assumptions about, for example, light conditions and head pose, or noiseless background in audio registrations, are met, but these are hardly compatible with real-life conditions. Although some methods are able to recognize emotional expressions across individuals, and to evaluate expressions over time (even though the considered time scale is still limited, and long lasting states such as moods cannot therefore be detected), context is usually not taken into account in current affective detection systems, which significantly limits the ability of such systems to reliably distinguish among ambiguous expressions. However, progress is being made very rapidly in this field, and more and more reliable emotion recognition systems are to be expected in a not too far future. As some of the examples provided in this section suggest, application scenarios that would benefit from these technologies are numerous, including enhanced human-computer interaction, tutoring systems, entertainment, assistive contexts; moreover, such systems could provide valuable support for emotion research in other disciplines, such as psychology and sociology, for instance by replacing long and tedious sessions of manual annotation with quick, automatic classification of facial actions or emotional content in speech.

7.3.2 Introducing machine emotion

Ideally, we can identify three perspectives that can inform the design of a system endowed with "own" emotions: the system might be able to express emotions, without actually having any corresponding internal state, or having one whose only purpose is to generate the outward expression; it might possess internal "emotional modules" that are used to complement and influence other cognitive operations (e.g. planning), but whose states are not communi-

cated to the outside world; or it might both have affective states that support its intelligent behavior, and actively take actions to express these states to an interacting user, or to other collaborating entities. The first approach is characteristics of systems for human-machine interaction, whereas the second one is more typical of artificial intelligence; the third approach combines both aspects (the communicative one, and the one related to affect-modulated reasoning) and might be particularly appropriate for robots that must cooperate with humans to perform a collaborative activity. In this section we will see some examples of affective systems inspired by these perspectives.

A wealth of *social robots* have been designed (Fong et al., 2003) in recent years, some of them with the exclusive aim of engaging human users in interaction, some other conceived for performing specific activities that require contact with humans, in situations where displaying appropriate social behavior might promote the acceptance of the robot, and a more effective collaboration with it. In both cases, exhibiting some form of emotional behavior is clearly a desirable feature: to this end, a robot (or an avatar) can take advantage of the same expressive channels we have already mentioned for humans: mainly, facial expressions, speech prosody, and body gestures.

Facial expressions are produced by FEELIX (Canamero and Fredslund, 2001), a humanoid LEGO robot built specifically for interacting with users; expressions for the basic emotions are created by changing the configuration of the eyebrows, and the lips. Expressions are triggered by tactile input on the robot's feet, and they depend on both duration and intensity of presses: for instance, a sustained high stimulation level results in an expression of anger. Facial expressions can be displayed also by virtual agents, such as avatars: in (Fabri et al., 2004) the use of emotional expressions in user avatars is investigated in the context of collaborative virtual environments, like virtual classrooms, in order to retrieve some of the emotional content typical of human live interactions that often gets lost when communicating at a distance. Whereas in these two examples emotional behavior gets displayed for the sole purpose of conveying an emotional state, in other cases the emotional expression can be functional in helping the robot perform a specific task that has been assigned to it, and that involves interacting and collaborating with people: this is the case for *service robots*, such as Minerva (Schulte et al., 1999), a tour-guide robot to be employed in museums. In order to attract people and guide them through its tour, Minerva uses facial expressions as a means of conveying intentions: for instance, if someone is standing in its way, preventing it from continuing its tour, it eventually shows an angry expression. In a similar spirit, Park et al. (2008) presented a system, both in virtual character and robot forms, that acts as an interface between a user and the multiple, sometimes complex devices composing a "smart house". The steward character exploits its ability to display facial expressions (the robot version uses lights and motions, instead) of emotions generated according to an OCC-inspired model, to make interactions more user-friendly; it is suggested that similar intelligent (and affective) interfaces to smart houses could be particularly suited for assisting the elderly, who might greatly benefit from the services offered by such an equipped residence, but require some

user-friendly support to manage all its sub-components in an effective way. In both this case, and in the tour-guide setting, showing emotions helps the robots successfully attend to their tasks, with higher satisfaction on the user's part.

Robots having the ability to convey emotional expressions have also been found to be useful tools in the treatment of children with autism (Ricks and Colton, 2010): as these children have trouble interacting with peers and caretakers, and find particular difficulties in deciphering emotional expressions, they can benefit from observing and trying to imitate the socio-emotional behavior shown by the robot, which has the advantage of being more predictable than human behaviors often are. FACE (Facial Automaton for Conveying Emotion – Pioggia et al., 2005) consists in a biomimetic face, endowed with an artificial muscular architecture and covered with artificial skin, that can reproduce realistic facial expressions for the six basic emotions; it has been proposed for use as an interface between children with autism and their therapist, who can manually select an appropriate expression to be displayed. FACE can be used to help the child gain familiarity with facial expressions of emotion, while he is guided by the therapist to name, match, and contextualize them.

Besides systems that employ the face channel for the expression of emotions, there have also been efforts toward synthesizing emotional speech. Cahn (1990) developed the Affect Editor, a system that produces appropriate speech prosody to pronounce a user-provided utterance with the desired affective connotation. This is achieved by varying parameters that model pitch, timing, voice quality, and articulation; a different set of parameter values is associated with each modelled emotional expression. In (Oudeyer, 2003) emotional speech for a robot is generated to resemble babbling in children (that is, produced utterances are meaningless) by specifying duration and pitch for each phoneme; a different set of prosodic parameters is used for reproducing each of five affective states (calm, happiness, sadness, anger, and comfort). Additional parameters were also introduced to allow for variations in the age of the robot, and intensity of expressed emotion. For both systems, validation experiments with human subjects, asked to name the emotion they perceived in the synthetic speech, confirmed that the intended emotional states were recognizable, although (as it is the case for human speech) accuracy was far from being perfect.

Robots and avatars that are able to express emotions by displaying a human-like facial expression, or by producing speech having an appropriate prosody, generally result to have a higher communicative power. However, even less anthropomorphic systems can succeed in conveying affective content by using body movements, posture, and orientation, as well as sounds and colors (Bethel and Murphy, 2008). An example is the tank robot described by Shimokawa and Sawaragi (2001): by using a genetic algorithm approach, it learns to perform movements that convey a particular emotion, among the basic ones, to the observer.

As we have seen for emotion recognition systems, emotion expression can be multimodal too. Kismet (Breazeal, 2002, 2003) is a social robot that engages users in interaction using facial expressions, emotional pseudo-speech, gaze direction, and approach/avoidance postures. Kismet possesses a motivation systems that included three drives: engage people,

play with toys, and rest. Drives that go unsatisfied for too long become pressing needs that generate emotional reactions and influence action selection; so do drives that enter an overwhelmed regime (for instance, when receiving aggressive and sustained stimulation from a user). For example, if Kismet is left alone for some time (its interaction drive is not satisfied), it expresses sorrow, which in turn is functional to attract the attentions of the user. Kismet's cognitive architecture is based on an appraisal theory of emotion, where situations are assigned an evaluation in terms of valence, arousal, and stance; the corresponding emotion intensity is computed from these values and, if a threshold is exceeded, the emotional expression is produced, by interpolation over the defined 3D emotion space.

Affective systems, especially when meant chiefly for interaction and entertainment, generally aim at achieving high *believability*: the Oz project (Bates et al., 1992) has specifically tackled this goal in building virtual worlds where characters endowed with cognitive and affective abilities live, and can interact with the user. The authors proposed an integrated cognitive architecture hosting an emotion module (Em) that interacts with other components, such as the module devoted to goal-directed behavior. The architecture of Em (Reilly and Bates, 1992) is inspired by the OCC model: emotions are generated, for instance, when evaluating the outcome of a relevant goal. Once generated, an emotion sets behavioral features, depending on the character's personality; activated features, in turn, initiate corresponding behaviors. In this, and in other systems described in this section, an appraisal theory of emotion has been used as inspiration for building an artificial emotional module. Indeed, computational models of human emotional processes (whether based on appraisal theories or not) as those presented in Section 7.2.3 can, in principle, be also employed to model believable affective agents; in fact, in some cases the distinction between explanatory models of emotions, and emotional architectures for artificial agents, does get somewhat fuzzy.

Another interesting and influential (Kismet's emotional module shares some of its aspects) emotional-cognitive architecture is Cathexis (Velásquez, 1997). This is a distributed model of affective phenomena, where each emotion family is represented by a dedicated system, called a proto-specialist. Each proto-specialist receives inputs regarding external and internal events, which can belong to any of four classes (neural, sensorimotor, motivational, and cognitive). Such inputs affect the intensity level for the considered emotion, to the point where a threshold is crossed, and the emotion is released; proto-specialists also have a saturation threshold, and a decay function, which can be set so as to model different temperaments. Although each proto-specialist operates on its own, it can also excite and inhibit other proto-specialists: for instance, the generation of happiness inhibits that of sadness; this mechanism also allows for mixed emotions to be modelled. Emotions, in turn, are one of the inputs to the behavior network, together with other motivations and events; the most highly activated behavior gets selected, and the associated emotional expression is produced. This model offers the possibility of designing emotional agents, such as the baby agent Simón presented in (Velásquez, 1997) as a testbed. An emotional architecture (EGO; see Arkin et al., 2003) is also included in the popular AIBO robot by Sony, a dog-shaped entertainment robot: behavior

selection depends on both external events, and internal drives generated by a homeostasis regulation rule that tries to keep the values of internal parameters (e.g. nourishment) within the desired range. The EGO architecture also allows for emotional symbol grounding: that is, when the robot encounters a new object, it learns its "meaning" by associating the object with the change in internal variables experienced when trying to execute a behavior on it (e.g. a new food source would be associated with the increased level of the nourishment variable registered after eating it).

Emotions are not only useful in interaction scenarios, to improve user-friendliness and believability of the interaction experience. They can also influence behaviors in a more general fashion, like they do in humans (e.g. experiencing fear triggers withdrawal behaviors), by prioritizing goals, redistributing attentional resources, or reinforcing learning processes. A model where emotions drive learning is described in (Gadanho and Hallam, 2001). Subjective feelings of the agent's internal state (such as hunger) elicit emotions, which in turn operate on feelings through the release of hormones (in a perception-emotion loop). Emotions, divided into positive and negative ones, also act as (positive or negative) rewards in a reinforcement learning task in which the robot must find food sources in its environment; if, for example, the robot has low energy and cannot find any food source, then sadness (negative reward) is elicited. Although the reported results on emotion-based learning are not conclusive, this model has the merit of explicitly considering the motivational role of emotion in learning adaptive behaviors. In (Lee-Johnson and Carnegie, 2010) emotions affect plan execution in a vehicle for indoor navigation, by either modifying in a smooth way its control parameters (reactive emotions), or by changing the probability to perform a given action (deliberative emotions). For instance, reactive fear, elicited when sensing a high density of obstacles on the way, modulates the velocity of the robot so that it can more safely pass through critical areas; happiness and sadness are used as (positive and negative) reinforcement values that get associated with nodes in a path, depending on whether this turned out to be a fast or a slow path. In this way, emotions can be employed to complement other cognitive processes (in this case, path planning) and support adaptive behavior. Indeed, it may be argued that this is exactly what emotions are actually about: adaptive behaviors that promote self-integrity and successful interaction with one's environment. What we call "emotion" might in fact be an emergent property of this basic adaptive mechanism. This was illustrated by the "Fungus eater" model proposed by Pfeifer (1994): given a very simple system based on associative learning, reflexes, and trade-offs between tasks related to self-preservation, behaviors eventually emerged that were judged, by human observers, to be affective behaviors, even though no emotion-related notions were ever modelled in the system.

In this section we have seen some examples of affective systems that express or model emotion; in so doing, we have also mentioned how these systems can be successfully employed in a range of application scenarios, such as entertainment, virtual tutoring programs, interactive platforms, and enhanced AI. However, there is still something that artificial systems still lack, and perhaps will never have: they cannot experience emotions – they can-

not *feel*. It could be argued that, to achieve that, computers should first develop consciousness. Even without going that far, the emotional experience in humans and animals is very much grounded in bodily changes (especially when assuming the Lange-Williams standpoint), which appear to be hard to reproduce in an artificial agent, especially when this is completely virtual. Still, some interesting investigations in this direction were made: in (Cañamero, 1997) simple virtual creatures, living in a 2D grid world where food, obstacles, and enemies are present, are implemented to reflect the development level of a very young creature. Perception and behavior are realized by means of dedicated modules (called agents); some agents record physiological variables, such as hormone levels and respiration rates, that define the creature's current bodily state. Emotions are interpreted as agents that, by releasing hormones, modulate the motivational and bodily states of the creature: when a behavior is selected in consequence of a drive (such as hunger), the triggering of an emotion by an external event can increase the intensity whereby the behavior is executed, or it can prevent the behavior from being executed at all; also, a different pattern of physiological variables will be obtained because of the hormone release, which results in the creature "sensing" a bodily change. This example shows the potentiality of taking into account low-level (i.e. physiological) aspects of emotion for explaining, and trying to reproduce, the modulatory effects of affective phenomena on high cognitive functions such as planning and decision-making.

7.4 Conclusion

The flourishing literature on emotion, both in the psychological/neuroscientific field, and in the Affective Computing one, witnesses the arising interest in better understanding, and possibly reproducing in artificial systems, affective phenomena and their influences on general cognitive abilities. In this chapter we have reviewed theories, findings, and models of emotional processes in natural systems: the resulting picture is still far from being clear yet, as little consensus can be found even for fundamental questions, such as the nature of emotion (e.g. which processes concur to form an emotion? how many emotion are there?) and its relationship with other affective phenomena, such as moods. However controversial, the body of knowledge on human and animal emotion built so far is nonetheless a precious source of inspiration for designing artificial systems that incorporate some form of emotional processing. The increasing enthusiasm for Affective Computing applications is a sign that the integration between "cold" cognition and affect is perceived to be useful and important for a range of objectives: on the one hand, more human-like intelligence can be achieved, by modelling the modulating influence of emotion on decision-making, memory, goal-directed behavior; on the other hand, more effective human-computer interaction systems, to be used in several application contexts (for example tutoring systems, telemedicine, or entertainment), can be developed if emotional information is processed together with more traditional communication channels. Therefore, the synergy between emotion research and computer science has already proved, although still in its early phases, to be fruitful, and worthy of further

exploration.

In this spirit we present, in the next chapter, a model for emotional interaction between a human and a machine (specifically, a pet robot), or between two synthetic agents (as those one could find in a video game). Human-machine interaction is usually based on an exchange of commands given by the user, and tasks performed by the machine. In an emotional interaction, *emotional states* are exchanged instead: that is, the messages being passed during interaction consist of information identifying the emotion experienced by an agent (either a human or an artificial agent). In order to achieve believable interaction, the artificial system does not have to simply mirror the emotion it has detected in the user; rather, it should respond with an appropriate, possibly different emotional state that depends, at the same time, on the emotional input from the user, on the history of the interaction, and on the specific characteristics of the system itself: that is, its personality and attitude. A model based on probabilistic finite state automata is proposed, so that the stochasticity of state transitions can provide a parallel to the relative unpredictability of human behavior. The robot's personality is encoded in the probability values for state transitions, which are subject to change during interaction to reflect the accumulated experience in dealing with the user (attitude formation). Different interaction patterns can therefore be expected when a user interacts with a different robot (that is, a robot with a different personality), or when he provides qualitatively different (e.g. positive vs negative) emotional inputs to the same robot. Several aspects involved in affective behavior are therefore addressed in this model: the role of personality in generating different interaction styles; attitude changes based on past experiences; the apparent unpredictability of emotional reactions. By taking these aspects into account, we argue that our model is able to generate natural (i.e. lifelike and credible) interactions, as they are inspired by the dynamics of human social interaction. An early version of this model was developed in the author's Master Thesis (Cattinelli, 2006), and subsequently expanded during the doctoral studies, in two main directions. First, a reinforcement learning approach was introduced in the agent-agent interaction setting to allow for automatic learning of an attitude that can consistently drive the emotional behavior of the interaction partner toward desired goal states: as a consequence, this mechanism supports autonomous adaptation to the interaction partner. Moreover, a set of quantitative analyses on interaction scenarios, based on Markov Chain theory, was developed: this provides statistical information on the expected outcome of having two specific agents interact, including the most frequent emotional states, which can also be used to validate the strategies discovered by reinforcement learning. The next chapter reports our published paper on this work[9] (Cattinelli et al., 2008).

[9]Notice that, for this reason, some of the introductory material covered here will be repeated in the next chapter as well; in this chapter, we tried to provide a more extensive, and updated, review of the related literature.

Chapter 8

A model for human-agent and agent-agent emotional interaction*

"If being human is not simply a matter of being born flesh and blood, if it is instead a way of thinking, acting and... feeling, then I am hopeful that one day I will discover my own humanity."
— Lt. Commander Data, 2338–2379

8.1 Introduction

Even though machines are increasingly spreading in every sector of our society, becoming indispensable tools able to solve everyday tasks, there are still many typical human abilities which cannot be reproduced by electronic devices: on the one hand human higher cognitive functions, such as language production and understanding, and, on the other hand, *emotional* functions: recognizing other people's emotions, reacting emotionally to situations, establishing relationships with an emotional content and so forth. The introduction of emotional components in computers can appear, at first, as pointless: the machine must be intelligent, not be able to feel emotions. But, if we aim to approach the ideal model of human intelligence, the emotional component cannot be ignored. Emotions are thought to be part of our decisional processes (Damasio, 1994), drive our learning, help self-preservation (fear for a dangerous phenomenon makes us move away from it), and are at the basis of human relationships. *Emotional intelligence* is an important part of all our intellective faculties.

The recently arisen effort in designing *emotional machines* which could understand, analyze, and synthesize emotions, derives from the acknowledged importance of emotions in human life. In the 1990s a new interdisciplinary research field (collecting contributions from computer science, neuroscience, psychology, sociology, and so forth) was proposed: *Affective Computing*, defined as "*computing that relates to, arises from, or deliberately influences emotions*" (Picard, 1997). The research field is, therefore, wide, but we can point out the following main themes:

- implementation of modules for human emotion recognition, based on physiological pa-

*The material reproduced in this chapter was published as (Cattinelli et al., 2008).

rameters (heart-beat rate, skin conductance, respiration, etc.) (Picard et al., 2001) or on *non-verbal communication* (Argyle, 1975) (facial expressions (Anderson and McOwan, 2006), posture (De Silva and Bianchi-Berthouze, 2004), gestures (De Silva et al., 2006), voice tone (Ciota, 2005));

- design of systems for simulating emotional states, which could communicate emotions readable by the human user (emotional avatars (Fabri et al., 1999; Lester et al., 2000));
- attempts at modeling emotional dynamics, to explain in formal terms how human emotional intelligence works and to reproduce this faculty in machines (Doshi and Gmytrasiewicz, 2004).

Combining modules for emotion recognition and production, emotional machines would be obtained, being able to interact with users not only by a limited set of standard commands, but also using emotion exchange as a more direct and natural communication channel. As a consequence, human-machine interaction would result easier and more effective, because it would be based on similar mechanisms as human-human relationships. Much work still remains to be done, though, since results in human emotion recognition are still quite unsatisfactory, due to the complexity of the task itself, and more generally to the difficulty in defining precisely what an emotion truly is. The same applies, of course, to emotion modeling.

In fact, emotions are still largely ununderstood, and different views have been proposed. According to (Scherer, 1984b), for instance, emotions are complex processes of which the external display is just one (and not necessarily the most important) of the components. Indeed, emotional expressions can be faked (this is the case for actors) or masked at some degree, so that they are not always valid cues for inferring someone's inner emotional state. However, external expressions are generally the only information we have on someone's emotional state, when interacting with them: blood pressure or skin conductance could be more faithful predictors of the actual experienced emotion, but measuring such data during the interaction would make it far less natural. Moreover, talking about emotions when referring to non-living entities like robots can be misleading, since the physiological changes naturally associated with the rise of an emotion cannot be reproduced (and this is particularly true for non-embodied agents); this is not of secondary importance, if we think that one traditional emotion theory (James, 1884) even states that bodily changes *cause* emotions to arise. Thus, in artificial intelligence or robotics emotional states are considered to be basically abstract, properly labeled (e.g. *happiness_state*) structures, upon which a set of behavioral responses is built (e.g. Kuhnlenz and Buss, 2004). Being our work focused on modeling emotional interaction in a robotic context, rather than human emotions per se, in what follows we will use the terms *emotion* and *emotional state* without pretense to address the whole complexity of these phenomena. Rather, we will focus on exterior emotional expressions and on general emotional categories which can be associated to them (in a very common-sense approach: "he is smiling" - facial expression - "then he must be happy" - inference about emotional state).

This work is, therefore, focused on analyzing and synthesizing emotional *behaviors*, rather

than human emotions as complex psychological processes. In this sense, related work can be traced in the wider field of behavior robotics. The work by (Chernova and Arkin, 2007) proposes a model for behavior selection in a QRIO robot, based on the robot's internal state and external inputs. To this purpose an activation level (AL) is associated to each behavior, defined as a weighted sum of four components. These describe, respectively, the robot's motivation and expected satisfaction, and the baseline activation and self-excitation associated to that behavior, plus a random noise parameter which adds variability in behavior selection. The AL formulation includes also a basis for robot's *personality* by means of a pair of weights, which can be set to facilitate self-centered or extroverted behaviors. Furthermore, the past history of performed behaviors is considered as a basis for autonomously learning routine sequential tasks. In (Inamura et al., 2004) robots learn a set of measurement-action pairs in an initially unknown environment; for instance, in an obstacle avoidance task, an adequate movement – right, left, forward – is found depending on the distance value reported by sensors. The measurement-action pairs are represented in a conditional probability table (CPT), which is updated through the interaction with the environment with the help of the information given by the user.

These works are mainly focused on action selection tasks, possibly helped by the interaction with users, rather than on human-robot interaction as such. The problem of modeling an emotional interaction which evolves through time may be better addressed through the use of a finite state automaton (FSA) (see, for instance, Hopcroft and Ullman, 1979). This model consists of states (which represent *emotional states*, such as happiness, or anger), inputs (events or information coming from the outside that are able to modify the emotional state) and a transition function, which describes the rules which transform the current state and the current input into a next state. Moreover, a "personality" of the agent could be defined and associated to the transition function, making agents with different personalities respond differently to the same stimulus. In fact, modeling emotional interactions requires taking into account individual variability, i.e. differences in characters and personalities which can affect the outcome of an interaction. Each individual has characteristic traits that should somehow be modeled in order to describe a likely interaction.

Since deterministic FSAs tend to produce stereotyped behaviors, a stochastic version of FSA, termed Probabilistic Finite State Automaton, PFSA, has been recently introduced in emotional interaction modeling. In a PFSA the transition function is stochastic: that is, given the current state and the current input, there are many possible next states, each entered with a given probability (Rabin, 1963; Paz, 1971). Indeed, the introduction of a stochastic component in an emotional interaction model leaves space for unexpected behavior (Chittaro and Serra, 2004; Kopecek, 2003; Kuhnlenz and Buss, 2004; Nomura, 1996): we do not expect that our interlocutor will always react the same way to the same situation, even if her personality remains unchanged.

In (Chittaro and Serra, 2004) an agent has a goal which can be accomplished by selecting different sequences of actions. Which action to perform is decided based on the agent's

personality that, in turn, determines the probability of each action: this selection process is represented as a PFSA. In (Kopecek, 2003) a PFSA is used as a personality model to describe the dynamics of a dialogue. Here, the entire automaton (that is, the ensemble of its inputs, outputs, states, and transition function) is referred to as the personality of the agent. By analyzing the dialogues, the possible automaton which generated them is searched. Both these models are prevalently static: that is, transition probabilities do not change according to the history of the interaction.

Another stochastic model, proposed as an emotional core for a robot, is a Hidden Markov Model (HMM), equipped in addition with input control (Kuhnlenz and Buss, 2004). Here, emotional states are the HMM's hidden states, to which different observable expressions are associated with different probabilities. Inner emotional dynamics is determined by a matrix of transition probabilities (not depending upon external stimuli), whose entries can be defined in order to design different robot personalities. The impact of different inputs on state transitions is coded into another matrix, which models emotional response to external events (perceptive information coming from sensors), as well as internal ones (produced by cognitive processes). Moreover, a forgetting filter is introduced which keeps a progressively decaying trace of past inputs, so that the probability for a state transition at a given time step is dependent not only on the events occurring at that time but also on past events, whose contributions are weighted decreasingly with the passing of time.

An interesting model for producing inter-individual relationships through conversations between individuals endowed with emotions and personality is proposed in (Nomura, 1996). Here, each individual is represented as an automaton, where inputs and outputs are actions (e.g. cooperation, or disregard), states are emotional states and personality is a parameter which determines the probability of each output as a function of the current emotional state. An additional parameter, the attitude, is also introduced, which modifies the probability of each next state, depending on the current input-output pair. So, both the transition and the output functions are parameterized, by attitude and personality, respectively. Attitude is a time-varying parameter: it is subject to updates based on the emotional state and the personality of the individual. While we share, as better explained in Section 8.2, the same keywords of attitude and personality, their meaning in (Nomura, 1996) and in our work is quite different (for a discussion, see Section 8.7).

In the following, we propose a novel, more complete model for emotional interaction between a human and an agent, or between two simulated agents, based on PFSAs: for each agent, states are its own emotional states, inputs are the emotions displayed by the interlocutor, while the transitions among states depend upon inputs and the agent's personality and attitude. Some transitions will be more probable in a friendly personality than in a crusty one, for instance. Moreover, transition probabilities are constantly updated throughout the emotional interaction depending on the agent's nature. A basic version of this model has been implemented in a real human-robot interaction. It is also shown that an adequate attitude can be acquired by an agent, simply through its emotional interaction with other entities. The

probabilistic features of the model, and especially the capability to adapt to the interlocutor (through the basic update rule based on the agent's nature or by reinforcement learning) can help improve interactions quality: the agent will change its attitude toward the interlocutor, in a dynamic way, depending on the input received, thus endowing the interaction with a more lifelike appearance. Moreover, we introduce here Markov chains to derive quantitative measurements of the expected behavior of two agents; as far as we know, this is the first attempt of a quantitative analysis of emotional interaction.

The paper is organized as follows: in Section 8.2 we introduce the basic model that can be applied to a human-robot interaction context. Section 8.3 proposes a reinforcement learning approach to obtain interactions aimed at particular goals. In Sections 8.4 and 8.5 we describe the implementation of the discussed ideas and the results obtained, and Section 8.6 develops a set of tools for a quantitative analysis of the expected behavior of the interacting agents. Lastly, Sections 8.7 and 8.8 summarize our work, further discussing the major features of the model.

8.2 Interaction Model

Our interaction model, describing the agent's emotional dynamics (for instance, let us consider a robot), is based on a probabilistic finite state automaton whose transition probabilities may change at each step. Formally, this is defined as a four-tuple $\langle S, U, P, s(0)\rangle$, where:

- $S = \{s_1, s_2, \ldots, s_N\}$ is the (finite) set of emotional states (e.g. happy, sad, angry, etc.) for the robot;
- $U = \{u_1, u_2, \ldots, u_M\}$ is the (finite) set of input, that is the emotions of the user (again, e.g. happy, sad, angry, etc.);
- $P = \{P_0, P_1, \ldots\}$ is the sequence of probabilistic transition functions: $P_t : S \times U \times S \to [0,1]$ for $t = 0, 1, \ldots$; and
- $s(0)$ is the initial state.

We explicitly notice that $\sum_{s' \in S} P_t(s, u, s') = 1$, for every t and every $(s, u) \in S \times U$. The sets of the robot's and user's emotional states can be defined freely, and they can consist of the same or of different elements. The only constraint is that the robot is able to reliably detect the user's emotional states u_j.

The robot reads the user's emotional state (for instance, by processing the video of her facial expressions), which becomes the input for its PFSA. At time t, based on the input, u_j, and on the current emotional state of the robot, the transition function P_t outputs the probability of entering any possible next emotional state. P_0 can be regarded as the robot's *personality*. We compiled several personality files, containing the probability for each triple (s, u, s') – where s is the current state, u the user input and s' the next state – to occur: robots with different personalities will tend to react differently to the same emotional stimulus.

The transition function changes as a function of time: P_t, called the robot's *attitude*, is updated depending on the robot's *nature*, which represents the "easiness" to reach certain subsets of emotional states. Nature is defined as follows. First, inputs are clustered in K different categories, c_k: nice, sad and bad inputs, for instance. For each category, a set of one or more target states is defined: $TS(c_k) = \{ts_j\}$. Moreover, each category is associated with an eligibility trace, which summarizes the inputs history (Sutton and Barto, 1998):

$$e_t(c_k) = \begin{cases} \alpha e_{t-1}(c_k) + h(c_k, u_j) & \text{if the current input is} \\ & \text{clustered in category } c_k \\ \alpha e_{t-1}(c_k) & \text{otherwise} \end{cases} \tag{8.1}$$

where α is the decay parameter and $h(c_k, u_j)$ represents the affinity between the current input, u_j, and the category, c_k: some inputs may be more representative of their category than others, and thus they will give a higher contribution to the relative eligibility trace.

When the trace associated with a category (say, c_k) reaches a predefined threshold value, the probability of entering all the target states for that category is incremented by Δ. Thus, for every target state $ts \in TS(c_k)$:

$$P_{t+1}(s, u, ts) = P_t(s, u, ts) + \Delta \tag{8.2}$$

The probability of entering the remaining states is decremented such that $\sum_{s' \in S} P(s, u, s') = 1$ for every $s \in S, u \in U$; this means that:

$$P_{t+1}(s, u, s') = P_t(s, u, s') - \frac{\Delta \cdot N_{TS}}{N - N_{TS}}, \ \forall s' \in (S \setminus TS(c_k)) \tag{8.3}$$

where N_{TS} is the number of target states for category c_k. So, if a robot has an imitative nature, the transition function will be changed so that the robot's behavior will tend to conform to that of the user: for instance, if the user has provided many positive inputs, the robot will more likely enter positive states; on the contrary, if the robot assumes a compensatory nature, its behavior will eventually diverge from that of the user.

The resulting model is a complete formalism for determining the agent's emotional response to the user's emotions, according to its key parameters of personality, attitude, and nature. In Figure 8.1 the main elements composing the model and the existing relationships among them are illustrated: here a more generic scenario is considered, where both interacting partners are artificial agents (as better explained in Section 8.4).

8.3 Learning Attitudes: a Reinforcement Learning Approach

Let us now suppose that our agent has a goal, for instance to make its user (frequently) happy: it needs to learn a behavior that allows it to reach such goal. This problem can be reformulated as a typical *reinforcement learning* problem (Sutton and Barto, 1998), where the agent learns a

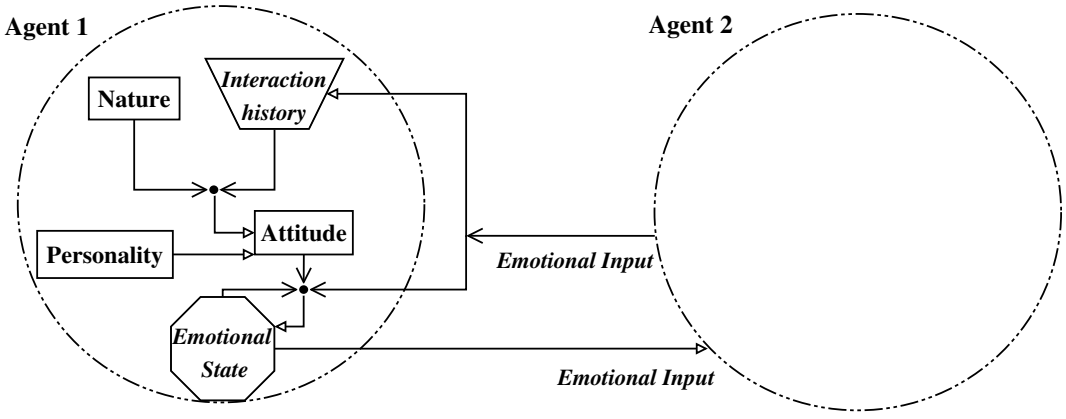


Figure 8.1: The key elements driving an emotional interaction in our model are shown. Here we consider two generic agents interacting: the same schema can be applied both to human-robot interaction, and to interaction between two synthetic agents. Arrows show dependences among the different parts of the model. For each agent, the emotional input coming from the other agent, together with its own current emotional state and attitude, determine the next emotional state, which is then output as an emotional input for the interacting partner. Attitude is initially coincident with personality, and successively modified during the interaction according to the input history and the agent's nature.

policy, that is a transition function, that maximizes the long-term reward obtained from the environment, represented here by the user. That is, the agent's behavior should lead the user most frequently into the subset of the desired states.

At each time t, the environment assumes an observable state s_t, which is, for our application, the user's emotional state. The agent chooses an action, a_t, among the possible ones, to be exerted on the environment. The action is the emotional state that the agent chooses to display to the user and it is a function of the actual state of the environment, s_t; such a function represents the agent's *policy* and it is defined by the stochastic function $\pi(s, a)$. Each action has a different effect on the user, who, in turns, changes her own emotional state to s_{t+1} and gives an instantaneous reward, r_{t+1}, to the agent. This reward can be positive or negative, according to whether s_{t+1} is or it is not a useful state in reaching the predefined goal; that is, the instantaneous reward will be positive if s_{t+1} belongs to the set of desired states (for instance, joyful states if the goal is to make the interacting partner happy).

The agent's optimal policy is the one that maximizes the long-term reward, R_t (expected discounted return, Sutton and Barto, 1998), that is

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \tag{8.4}$$

where γ is a discount rate and T is the final step of learning (which goes to infinite in case of infinite horizon problems, like in the present case).

One of the most effective techniques for learning the optimal policy is *Q-learning* (Watkins, 1989), where the agent learns an *action value function*, $Q(s, a)$, that gives the expected long-term

return starting from state s, executing action a and, from that on, following the given policy, $\pi(s,a)$. For every step of each learning episode, the function $Q(s,a)$, is updated according to

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \tag{8.5}$$

This technique allows the agent to learn the optimal value function and, at the same time, to learn the optimal policy for the given goal.

8.4 Implementation

The described model was implemented in an emotional interaction between a human and a robot and between two agents.

In the first case, emotional interaction between an AIBO™ robot[1] and its master has been developed. The set of inputs, U, consists of the six universal emotions according to Ekman (1992b): joy, sadness, surprise, anger, fear, and disgust, plus the neutral emotion. S is restricted to four states: neutral, joy, sadness, and anger, for sake of simplicity, and the neutral state is chosen as initial state, $s(0)$. The output of the robot, at each step of interaction, is a predefined sequence of body movements, sounds and patterns of lights that represent the current emotional state of the robot.

The master's emotions are detected through the analysis of her facial expressions. This has been accomplished processing the video stream transmitted by AIBO's camera to the on-board processor (MIPS R7000, 576 Mhz). Basic image processing techniques, as color segmentation, border extraction and block matching, have been implemented in order to meet the real-time response requirement (Campadelli and Lanzarotti, 2002). The image processing module identifies a set of expressive features (e.g. mouth corner or inner portion of the eyebrow), which are mapped onto Action Units. These are the elementary facial movements defined by the *Facial Action Coding System* (FACS (Ekman and Friesen, 1978), see also (Magnenat-Thalmann et al., 1988) for a similar approach), and are then mapped to emotion expressions, through a fuzzy-like system of recognition scores. A four-step interaction with AIBO is shown in Figure 8.2: in the left panels, AIBO's master displays her emotional state to AIBO through an adequate and well-defined facial expression. This is input to the emotional interaction model of AIBO, which, in turns, produces a new emotional state of AIBO, displayed through an adequate behavior as shown in the right panels.

In order to study more extensively the proposed interaction model, we have applied it to the interaction between two stochastic agents, where each interacting agent can be represented by a PFSA. In this case the state of the first automaton, A^1, becomes the input for the second one, A^2, and vice versa. Thus, we have two agents $A^1 = \langle S, U, P^1, s(0)^1 \rangle$ and $A^2 = \langle S, U, P^2, s(0)^2 \rangle$, where:

- the set of emotional states S is the same for both A^1 and A^2;

[1] Web site: *http://www.sony.net/Products/aibo*.

Figure 8.2: Four phases of an emotional interaction with AIBO. In the left panels, the emotional expression displayed by the master. In the right panels, the emotional response of AIBO. In particular, the sad expression is displayed by AIBO lowering its head and playing a sad melody; in the angry expression AIBO moves quickly forward in an aggressive fashion and growls, while in the happy expression the robot wags its tail and barks happily. In the neutral expression AIBO stands still, looking around as if it is uninterested in the user.

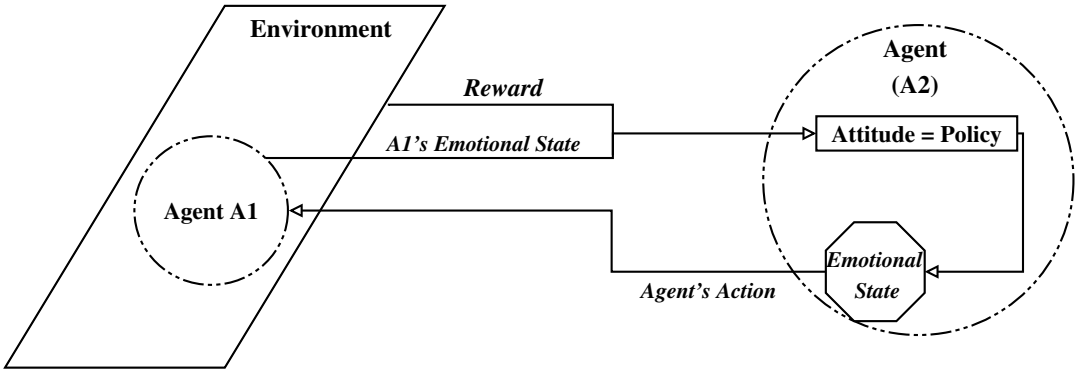


Figure 8.3: The reinforcement learning approach applied to our interaction model. Here, one agent (namely, A^2) acts as the learning agent, while the other one (namely, A^1) embodies the environment. Actions exerted by A^2, that is its own emotional states as externally displayed, cause A^1 to enter a new emotional state. If this is a goal state, a positive reward is delivered to the learning agent. The reinforcement learning algorithm will then update the agent's policy (which, in our terminology, is its attitude), which in turn will provide a new action to be executed on the environment.

- the set of possible inputs, U, is coincident with the possible states, S;
- the probabilistic transition functions, P_0^1 and P_0^2, are different at start, that is the two agents have different personalities;
- the initial states $s(0)^1$ and $s(0)^2$ are different.

We have extended the set of possible emotional states for the two automata, S, by including, for each basic emotion, three different levels of intensity. Thus, the emotion of anger, for instance, is now represented by three different states, corresponding to low, medium, and high anger intensity: ANNOYED, ANGRY, and FURIOUS, respectively. A total of $N = 19$ states results, including the neutral emotional state.

Emotional inputs are clustered into M categories. In the following six categories are considered, each associated with a different basic emotion (so we have, for instance, the joy category, which is associated with all joyful inputs). Each level of an emotion contributes differently to the corresponding category eligibility trace ($h(c_k, u_j)$ in Eq. 8.1), so that probability update can be triggered by few intense inputs or many consecutive low-level inputs.

We first used a simplified version of the above-described model to study the interaction cycles with the aim to determine the most frequent behavioral patterns for the personality of the two agents. To the scope, we assumed that A^2 can be described as a deterministic stationary automaton and we analyzed the behavior of A^1 under the hypothesis that its transition probabilities, which define its personality, are not modified during the emotional interaction, making the automaton A^1 stochastic but stationary. Under these assumptions, we could adopt an algorithm based on depth-first search on the computation tree (Cormen et al., 2001) to extract the probability of each interaction cycle, that is a sequence of emotional states of A^1 which starts and ends in the same state.

Afterwards, we used the full model to analyze if a successful emotional relationship can be discovered by an agent, without any a priori information. To this end, let us regard A^1 as a probabilistic stationary environment for A^2: the emotional states output by A^1 are directly observable by the learning agent, A^2. A^2 has to learn, through reinforcement, a policy such that it obtains the maximum possible reward from A^1; that is, it has to learn a policy that outputs a set of actions, which let A^1 enter the predefined goal states most often (see Section 8.5). We have chosen to give the same amount of reward (namely, $r = 1$) whenever one of the goal states is reached; therefore, if the goal is, for instance, to make A^1 sad, each time it enters any state in the goal set, i.e. {MELANCHOLIC, SAD, IN_DESPAIR}, the same reward r will be delivered to A^2. After learning has been completed, the final policy defines a set of transition probabilities, which represent the attitude of A^2 after it has been adapted to the personality of A^1. Figure 8.3 summarizes the implemented reinforcement learning scenario.

8.5 Results

In the case of interaction between two agents, we could observe very different behavioral patterns, depending on their personality and nature, and possibly on the goal set. Let us start by briefly discussing how the key concepts of personality, attitude, and nature contribute to shaping an interaction.

In the simplest case, agents do not experiment any attitude evolution over time; therefore, the interaction depends only on the personality of the two agents. An evaluation of an agent's personality can be attained by analyzing its transition matrix, that is the three-dimensional matrix consisting of the transition function value for each possible triple (s, u, s'). Personalities can be designed by stressing, by means of a high probability value, the relevance of particular transitions of interest. For instance, a friendly personality can be characterized by high probability of entering positive emotional states (i.e., joyful), mixed with mirror behaviors (e.g. being sad if the partner is sad) as a sign of emotional involvement.

When the agents' attitude is allowed to evolve with time, the history of past inputs drives their interaction in a direction that is determined by the agents' nature. To illustrate this point, let us consider an interaction setting where both agents have a friendly personality (i.e., their transition matrices were crafted to define a friendly personality). If both agents are endowed with an imitative nature, their emotional relationship quickly converges to a sequence of positive emotional states, since the inputs that each agent is observing are most of the time positive. On the contrary, by setting one of the two agents' nature to compensatory, negative emotional states do occur as the emotional interaction goes on: upon receiving mostly positive inputs, the compensatory agent will, in fact, increase its probability of entering negative emotional states.

Having agents with quite different personalities interact also helps in obtaining more dynamic interactions. For instance, in Figure 8.4, we had agent A^1, endowed with a friendly personality, interact with A^2, whose personality was obtained by linearly combining, with equal

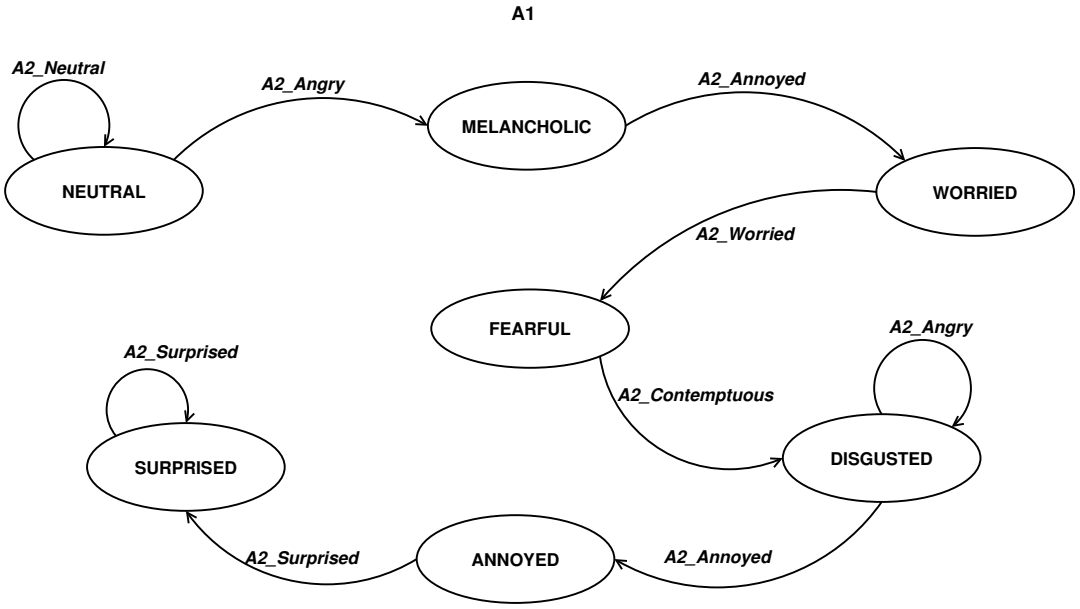


Figure 8.4: This state transition graph shows 10 steps of an interaction between a friendly agent A^1 and an agent A^2, whose personality is obtained from the friendly one perturbing it with random traits. Only A^1 transitions are shown, while A^2 transitions can be easily derived from the arc labels. This interaction is rather dynamic, as it can be seen by the variety of states entered by the two agents; there is not a quick convergence to positive emotional states as we would expect if both agents had the same personality.

weights, the transition probability matrix for the friendly personality with one, randomly generated, describing deterministic transitions. In this case more complex interaction patterns tend to emerge, with a variety of experienced emotional states. Therefore, by carefully tuning personalities and natures one can obtain interactions with the desired characteristics.

Alternatively, the overall trend of an interaction can be predetermined by giving a goal and letting the agent learn by itself the most adequate policy to reach that goal, during the interaction with another agent. Reinforcement learning is used here to this scope. A few results are now presented.

Let us suppose that at a certain time, t, the goal of making its friendly partner, A^1, angry is assigned to agent A^2. This means setting the states ANNOYED, ANGRY, and FURIOUS as goal states. Through Q-learning (Watkins, 1989) A^2 does learn a new policy (that is, it changes its attitude), to accomplish this goal, as shown in Figure 8.5 where 100 steps of emotional interaction between A^1 and A^2 are reported, after learning has been completed. As it can be appreciated, 78% of the states reached by A^1 are goal states, showing that A^2 did learn a policy effective for the goal. Few state transitions occurred frequently during the interaction: in particular, the cycle on the ANGRY state was repeated 50 times over 100. This experimental observation is confirmed by the theoretical analysis of the cycle probability carried out on the 3D transition probability matrix: for instance, the four-step-length cycle ANGRY-ANGRY-ANGRY-ANGRY has, alone, a high probability of occurring (0.58) in this particular setting.

The history of a 100-step interaction between the same agents A^1 and A^2, when the goal

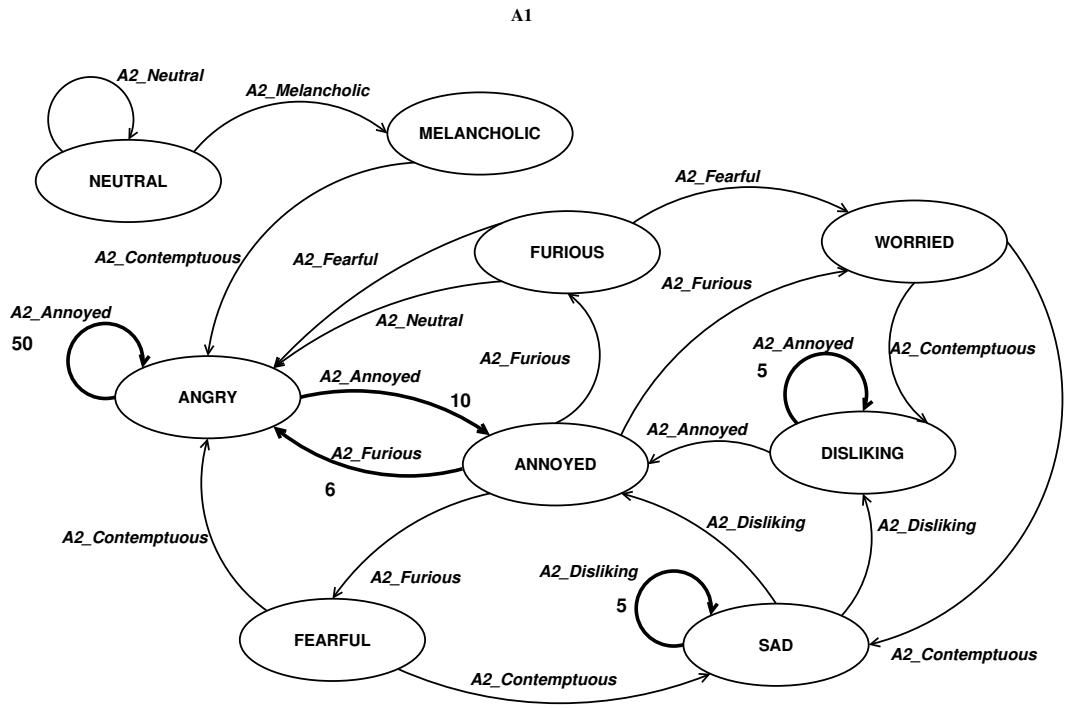


Figure 8.5: The interaction between a stochastic agent, A^1, with a friendly personality and a second stochastic learning agent, A^2, is described in this state transition graph. The attitude of A^2 was modified with the goal to make A^1 angry most of the time. A total of 100 interaction steps are reported after learning has been completed. The states visited by A^1 are written inside the graph nodes, while the arcs represent the transitions to next states; each transition was induced by the action chosen by A^2, which is written besides the arc. Notice that, due to the stochastic nature of A^1, different next states can be reached from the same actual state with the same action (e.g. SAD/A2_DISLIKING). The starting emotional state is NEUTRAL. The number of occurrence of the most frequent transitions is reported in bold besides the corresponding arc.

for A^2 was to make A^1 surprised (the target states are now WONDERING, SURPRISED, ASTONISHED) is reported in the graph of Figure 8.6. In this case, the rate of goal states entered by A^1 was 95%, with the most frequently occurred cycle on the SURPRISED state.

Even though, in the two examples shown, A^1 has the same personality (and therefore, identical transition probabilities), the actions performed by A^2 according to the two different learned policies are effective in driving the emotional interaction with A^1 to very different groups of states as prescribed by the different goals set, thus producing very dissimilar interaction patterns. Given the same environment, having different goals necessarily leads to different policies.

On the other hand, given the same goal, but a different environment to act in, different policies will be developed by the learning agent. For instance, we considered a scenario where A^1's personality was suddenly changed, while A^2 maintained the attitude that it had previously learned, with the goal of making A^1 angry. In this situation, A^2's policy became less effective in reaching the goal set, since that policy was learned based on a different en-

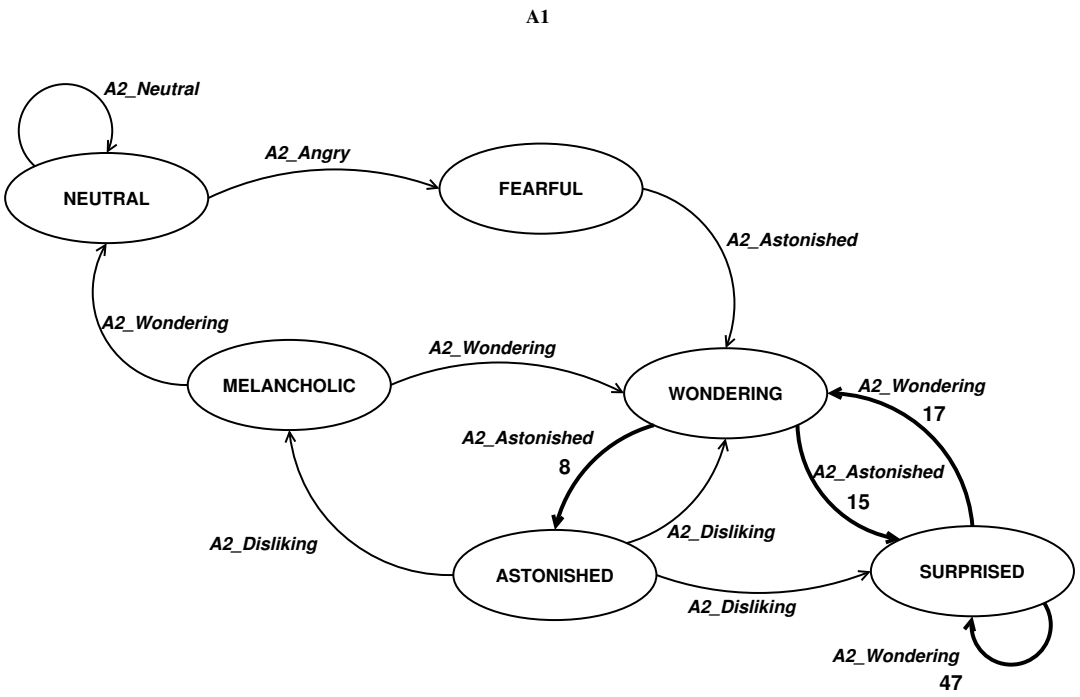


Figure 8.6: The interaction between the same friendly stochastic agent, A^1, of Figure 8.5 and a second stochastic agent, A^2, with a different goal, that is to make A^1 surprised. A total of 100 interaction steps are reported after learning has been completed. The number of occurrence of the most frequent transitions is reported in bold besides the corresponding arc.

vironment (i.e., a different A^1). After A^1's personality was changed, the interaction showed many different transitions, each of which occurred infrequently (Fig. 8.7), rather than the few transitions occurring rather frequently shown in Figures 8.5 and 8.6.

Figure 8.8 shows the rate of goal states, p_{goal}, reached by A^1 over time, referred to blocks of $1,000$ Q-learning iterations. Initially, when A^2 explores the state-action pairs, p_{goal} oscillates. It starts to increase when the agent discovers an effective policy, around the 40^{th} block, to plateau at 70% at the 100^{th} block, when the policy of A^2 does not receive any meaningful update anymore. At this stage, the agent ceases to explore, through random actions, the state-action space and just follows the learned policy. Using this policy, A^2 was able to obtain consistently the desired behavior from A^1, as shown by the high rate of goal states reached.

At the time corresponding to the 110^{th} block, the personality of A^1 was changed. A random component was added to the state transition matrix, to obtain a different personality of A^1. The new personality was obtained by linearly combining P_0^1 with a random transition matrix, with blending coefficient equal to 0.5. As a result the personality of the agent remained basically friendly. Nevertheless, the success rate abruptly decreased to around 20% and the agent A^2 had to learn again to cope with the new A^1. At this point, Q-learning was called to operate. As we can see in Fig. 8.8, after a set of 10 blocks, the goal rate starts increasing to plateau around the 220^{th} block at a success rate of about 51%, which is lower than with the previous A^1's personality. A sequence of 100 interaction steps under these conditions (A^1 has

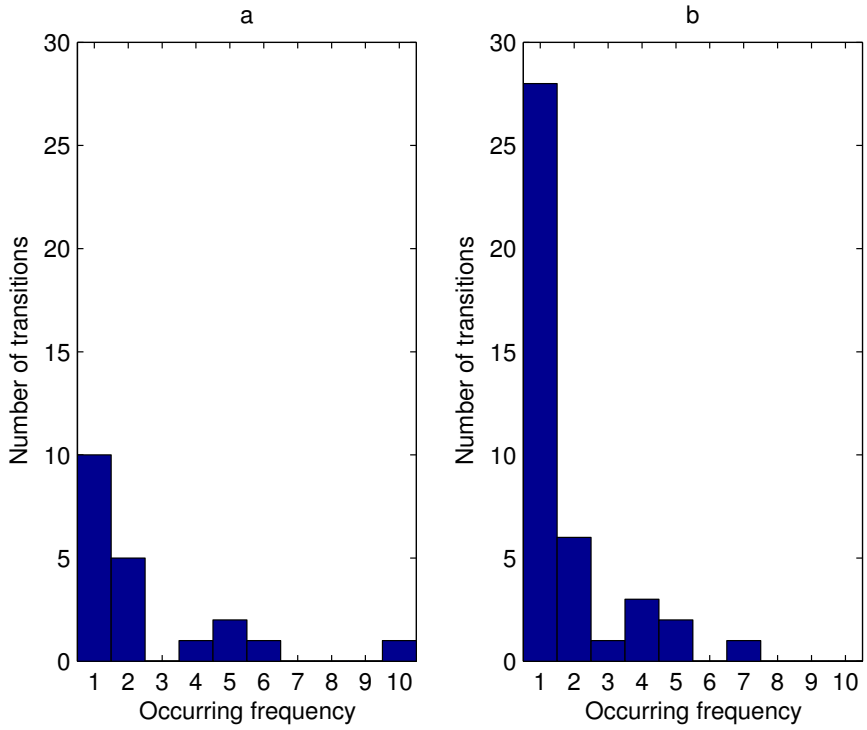


Figure 8.7: Panel a shows the histogram of the state transitions, that is of the triples (s, u, s'), for the 100 emotional interaction steps, whose state transition graph in shown in Figure 8.5. The cycle on the ANGRY state alone occurred 50 times and it is not shown in the histogram; as it can be seen all the other state transitions occurred rather infrequently. Panel b shows the histogram of the state transitions for 100 steps occurred just after the personality of A^1 was changed, by introducing some random traits in the friendly personality. The policy of A^2 remained that previously learned with the goal to make A^1, with the previous personality, angry. In this case, excluding one state transition (permanence in state DISLIKING on input ANNOYED) which occurred 28 times (also not shown in the histogram), all the other transitions occurred a low number of times (mainly, only once).

a new personality, A^2 learned a new policy for it) is shown in Figure 8.9. The state transition graph shows several frequent transitions which involve non-goal states. Comparing these results with the goal rates obtained for A^1's previous personality, the agent A^2 was less able to interact successfully with the agent A^1, when endowed with this new personality.

As a closing remark, we observe that the actual interaction resulting from the learning process can assume very different shapes: these depend not only on which goal has to be accomplished, but also on the dynamics of the learning process itself. In particular, the end result is influenced by the starting policy (A^2's personality) and by the stochastic nature of the environment. Together, these two aspects determine which regions of the state-action space are explored. In fact, A^2's starting policy determine which actions and how often they will be tried during learning. This means that some actions, though in principle useful in reaching the goal, will produce little or no reward at all, simply because they will hardly be experimented. We can think of someone who, although his goal would probably be accomplished by acting a

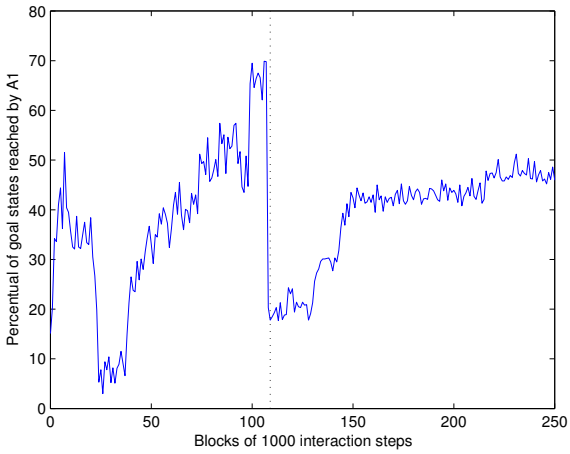


Figure 8.8: Rate of goal states entered by A^1 during learning, computed over blocks of 1000 interaction steps. At the beginning of block 110^{th} A^1's personality is changed, which produces a sharp decrease in the goal rate, until a new effective policy is discovered.

certain way, still is unwilling to carry out these actions because of his personality. Moreover, since A^1's responses to A^2's actions are not deterministic, different reinforcement learning runs will result in different policies, depending on which state-action pairs will be observed. For instance, in the interaction instance shown in Figure 8.6 A2_SURPRISED would be effective in keeping A^1 in the SURPRISED state (thus, a goal state); however, it is never performed since the reinforcement learning process was unable to highlight that particular state-action pair, preferring A1_SURPRISED-A2_WONDERING instead.

In summary, we showed how an emotional interaction can be driven towards predefined directions by using two alternative ways. The first way involves the tuning of interacting personalities and natures, in order to obtain the desired interaction: this could be, for instance, characterized by mostly positive states, or on the contrary by a wider variety of states. Alternatively, reinforcement learning can be applied to have one agent autonomously learn how to get the desired behavior from the interacting partner.

8.6 Quantitative behavior analysis

In a probabilistic framework like the one described here, each different run of the application will produce a different interaction pattern. Therefore, apparently there is no easy way to predict the outcome of an interaction. However, we show here that, if we take a picture of the two agents' attitude at some point in time (for instance after the adaptation/learning process is concluded), their asymptotic behavior can be described by resorting to Markov chains theory. To this purpose, we first summarize some well-known properties of (homogeneous finite) Markov chains that we are going to apply in our context (see for instance Iosifescu, 1980; Häggström, 2002).

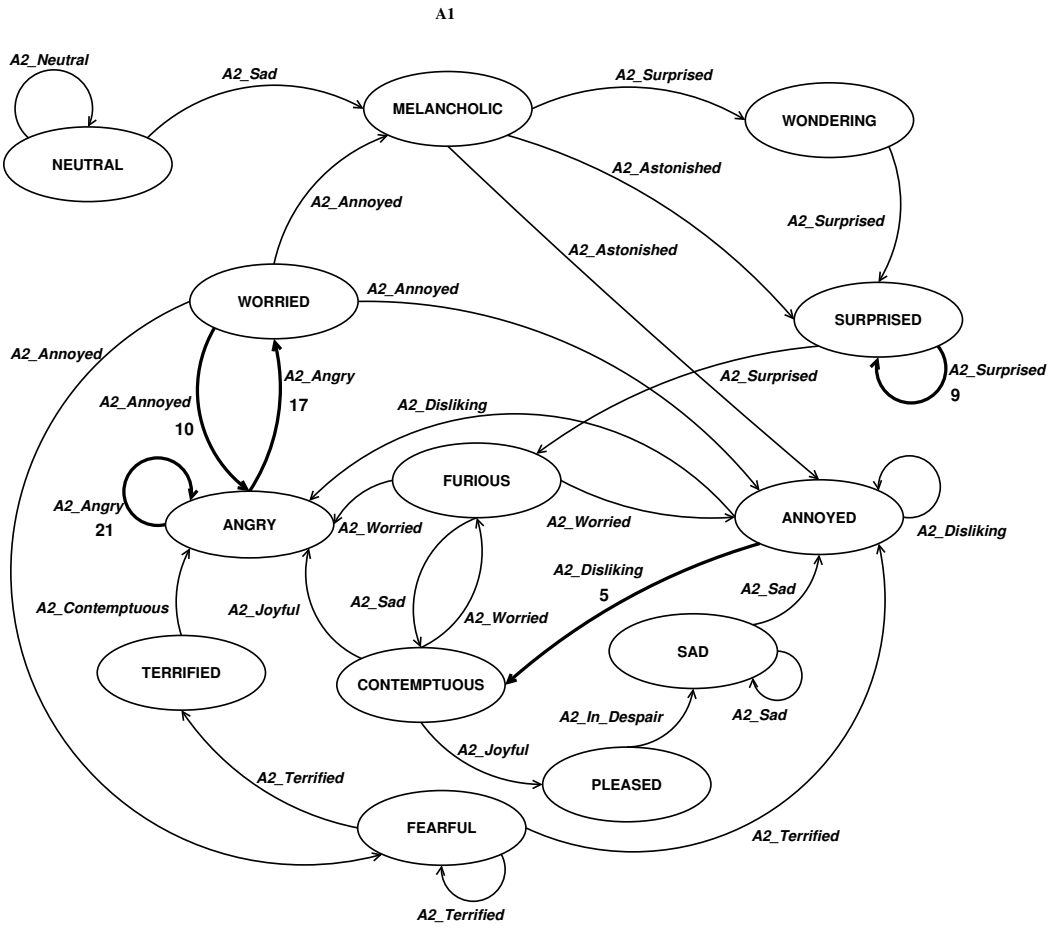


Figure 8.9: An instance of interaction between a stochastic agent A^1, whose personality, initially friendly, was changed during interaction, and a second stochastic agent, A^2. A^2 had to learn a new policy to adapt to the change in personality of A^1, while maintaining the same goal, namely to make A^1 mostly angry. A total of 100 interaction steps are shown.

8.6.1 Markov chains theory

Given a finite set S and a stochastic matrix $P = [P_{ij}]_{i,j\in S}$, i.e. a matrix satisfying conditions $P_{ij} \geq 0$ and $\sum_{l\in S} P_{il} = 1$ for all $i, j \in S$, let $\{X_n\}_{n\geq 0}$ be a Markov chain taking values on the set of states S with transition matrix P. Moreover, for every $n \in \mathbb{N}$, let $\mu^{(n)}$ be the probability distribution of X_n, which we consider here as a column array with index in S. Then, the values of $\mu^{(n)}$ and the n-steps transition probabilities between states can be computed from the powers of the matrix P: for each n and every $i, j \in S$, we have

$$\Pr(X_n = j \mid X_0 = i) \;=\; (P^n)_{ij} \tag{8.6}$$

$$\mu_j^{(n)} \;=\; \Pr(X_n = j) \;=\; (\mu^{(0)\prime} P^n)_j \tag{8.7}$$

The properties of $\{X_n\}_{n\geq 0}$ are of particular interest in the case when the matrix P is *primitive*, that is $P^k > 0$ for some $k \in \mathbb{N}$ (i.e., $(P^k)_{ij} > 0$ for all $i, j \in S$). This is equivalent to require that P is irreducible (meaning that the corresponding graph of transitions with non-null probability is strongly connected) and aperiodic (i.e., the greatest common divisor of the lengths of cycles is 1). If P is primitive then the following properties hold, which either are classical results or can be easily derived from standard issues:

1. There exists a unique probability distribution π over S such that

$$\pi' \, P \; = \pi' \tag{8.8}$$

which is called *stationary distribution* of the chain. Note that π' is a left eigenvector of P corresponding to the eigenvalue 1. This means that, if π is the probability distribution of X_0, then $\mu^{(n)} = \pi$ for all n.

2. For every $i, j \in S$

$$\lim_{n\to+\infty} (P^n)_{ij} \; = \; \lim_{n\to+\infty} \Pr(X_n = j) \; = \; \pi_j \tag{8.9}$$

that is, for large n, the probability that X_n equals j approximates π_j independently of the value of X_0, that is the initial state of the chain.

3. The values of π are related to the average waiting time for the first entrance in the states on the chain. More precisely, for every $i \in S$, let τ_j be the random variable defined by $\tau_j = \min\{n > 0 \mid X_n = j\}$. Thus, for all $i, j \in S$, the value $E_i(\tau_j) = E(\tau_j \mid X_0 = i)$ is the mean waiting time for the first entrance in j starting from the state i. Then, it turns out that

$$E_j(\tau_j) = 1/\pi_j \text{ for each } j \in S \tag{8.10}$$

4. For $i \neq j$, the values $E_i(\tau_j)$ can be computed as follows. Let $G(z)$ be the matrix of polynomials in the variable z given by $G(z) = I - Pz$ and denote by $r_{ij}(z)$ the entry of indexes i, j of the adjunct of $G(z)$, i.e. $r_{ij}(z) = (-1)^{i+j}\det(G_{ji}(z))$ where $G_{ji}(z)$ is the matrix obtained from $G(z)$ by deleting the j-th row and the i-th column. Then, through the computation of some derivatives, one can prove that

$$E_i(\tau_j) \; = \; \frac{r'_{ij} r_{jj} - r_{ij} r'_{jj}}{r_{jj}^2} \tag{8.11}$$

where $r_{ij} = r_{ij}(1)$, $r_{jj} = r_{jj}(1)$, $r'_{ij} = r'_{ij}(1)$ and $r'_{jj} = r'_{jj}(1)$ (which are all well defined in this case).

5. One can also evaluate the error in the approximation of $\mu^{(n)}$ towards π. To this end, let us recall that the total variation distance between two probability distributions μ, ν

defined on the same finite set S, is given by

$$d_{TV}(\mu,\nu) \;=\; \frac{1}{2}\sum_{i\in S}|\mu_i-\nu_i| \tag{8.12}$$

For such a distance the following relation is satisfied: $d_{TV}(\mu,\nu)=\max\{|\mu(A)-\nu(A)|\mid A\subseteq S\}\leq 1$, and hence it yields a complete evaluation of the difference between two (finite) probability measures. Moreover, for every stochastic matrix T on S, we define the coefficient

$$m(T) \;=\; \frac{1}{2}\max_{i,j\in S}\{\sum_{l\in S}|T_{il}-T_{jl}|\} \tag{8.13}$$

which is the maximum (total variation) distance between pairs of rows of T. Then (still assuming P primitive) for every $\varepsilon>0$ we have

$$d_{TV}(\mu^{(n)},\pi) \;\leq\; \varepsilon \tag{8.14}$$

for all $n\in\mathbb{N}$ such that

$$n \;\geq\; t\left(1+\frac{\log_2 k-\log_2\varepsilon-1}{-\log_2 m(P^t)}\right) \tag{8.15}$$

where k is the cardinality of S and t is the smallest integer such that $P^t>0$.

8.6.2 Markov chains for interaction analysis

Let us now go back to our interaction model and explicitly notice that two interacting agents can be regarded as a single closed system, as transition probabilities do not depend upon external input; here, states are pairs of emotional states (one for each agent). Such a system can be, therefore, described by a single matrix, M, collecting transition probabilities over its states: if S is the set of emotional states for both A^1 and A^2, the set of states over which M is defined will be $S\times S$. In other words, $M(i,j)$ represents the probability for the system to go, in one step, from state $i=(a,b)$ to state $j=(a',b')$, where a,a' are emotional states for agent A^1, and b,b' are emotional states for agent A^2. Matrix M can be easily derived from the two agents' personalities: entry $M(i,j)$ is obtained by multiplying $P^1(a,b,a')$ and $P^2(b,a',b')$.

In our context it often turns out that M is not an irreducible matrix; hence we cannot immediately apply the results presented in Section 8.6.1. M usually consists of more than one strongly connected component, but among these we can focus on *essential* components: these are defined as strongly connected subgraphs that cannot be left once entered. Therefore, at some point in the interaction the system enters one of these components, and afterwards only its states are visited. On the other hand, nonessential components are transient, and with probability 1 they will at some point be abandoned and never visited again; this is the reason why they can be excluded from our analysis.

In all the examples we considered, we found only one essential (strongly connected) component of M, which turned out to be aperiodic, too. Let us call M_{red} the transition matrix for such essential component. Its stationary distribution π can be computed as shown in Eq. 8.8. In our framework, π_i will thus provide the probability for the system of being in a state i, which represents a pair of emotional states, one for each agent. The corresponding probability for each agent separately can then be derived, by summing the probabilities over the states for the other agent:

$$\pi_1(a) = \sum_{b \in S} \pi_{(a,b)} \tag{8.16}$$

$$\pi_2(b) = \sum_{a \in S} \pi_{(a,b)} \tag{8.17}$$

We observe that, by using standard methods (Iosifescu, 1980, chap. 3), one can compute the average waiting time required by the system to be absorbed into the essential component (also called time to absorption). In all our experiments, the average time to absorption is smaller than 3. This justifies our choice to study a restriction of M to the set of essential states.

The stationary distribution provides a description of our (reduced) system yielding the limit probability of visiting each state i. By comparing the stationary distribution for two different systems, i.e. two systems consisting of different interacting agents, we can quantify how their behavior differs: in particular, different interaction scenarios will be characterized by different states having maximum probability to be visited. A special case occurs when the two systems being compared are the system at the beginning of the interaction (before any adaptation occurs) and the system resulting from the adaptation process at the end of that interaction. In this particular case, by comparing the stationary distributions found for the two systems we can quantify the results produced by the interaction itself; here we are interested in identifying the most probable states, before and after the evolution of the two agents' attitude.

Let us start with a simplified example, where the set of emotional states includes just the six basic emotions plus the neutral state. In this setting, a *random* agent A^1 was designed, for which every transition (s, u, s') has the same probability. The transition matrix M can be computed assuming A^1 interacts with a friendly agent A^2 without adaptation. Twelve strongly connected components were found, among which only one was identified as an essential component of 38 states (over a total number of 49 states). For this component, we computed the stationary distribution, and we found that the most probable states of the system are (JOYFUL, JOYFUL) ($p = 0.1144$) and (SAD, SAD) ($p = 0.1118$). Focusing on A^2's states only (A^1 has equal probability of entering each state, by definition), we found this stationary distribution: $\pi_2(\text{SAD}) = 0.3573$, $\pi_2(\text{JOYFUL}) = 0.1906$, $\pi_2(\text{NEUTRAL}) = 0.1857$, $\pi_2(\text{FEARFUL}) = 0.1435$, $\pi_2(\text{ANGRY}) = 0.0660$, $\pi_2(\text{SURPRISED}) = 0.0548$, $\pi_2(\text{DISGUSTED}) = 0.0022$.

We then had A^1 adapt to A^2, setting for it an imitative nature. As a result, we expected that the stationary distribution over A^1's states would somehow approximate that found for A^2 at the beginning of the interaction. This was confirmed by our experimental analysis: after

200 steps of interaction, we found that A^1's stationary distribution was: $\pi_1(\text{SAD}) = 0.3413$, $\pi_1(\text{JOYFUL}) = 0.1547$, $\pi_1(\text{FEARFUL}) = 0.1430$, $\pi_1(\text{ANGRY}) = 0.1080$, $\pi_1(\text{SURPRISED}) = 0.0847$, $\pi_1(\text{DISGUSTED}) = 0.0847$, $\pi_1(\text{NEUTRAL}) = 0.0837$. In particular, we can see that the most probable state is SAD, as it was for A^2 at the beginning of the interaction. This happens because, when an imitative nature is set, the adapting agent increases its probability of entering those states which have been observed more often in the other agent.

Similarly, we studied the asymptotic behavior of the systems depicted in Figs. 8.5, 8.6, and 8.9. For clarity purposes, we will refer to these systems as *System1*, *System2*, and *System3*, respectively. In these examples, the total number of states of the system is $19^2 = 361$.

Figure 8.5 shows an instance of interaction between a friendly agent and a second agent having learned how to make A^1 angry most of the time. The corresponding matrix M contains only one essential component of 15 states; therefore we could study the reduced matrix M_{red}, of size 15×15. The computation of the stationary distribution for this component showed that the most probable state is (ANGRY, ANNOYED), with $p = 0.5148$, followed by (ANNOYED, FURIOUS), with $p = 0.1548$, and (SAD, DISLIKING), with $p = 0.0973$. This is mirrored by the actual behavior of the system as shown in Fig. 8.5: for instance, we can see that, over 100 steps, the state pair (ANGRY, ANNOYED) occurred 60 times, thus rather close to the theoretically predicted frequency.

Similarly, when we considered *System2*, consisting of a friendly agent interacting with an agent trained for making the partner surprised (see Fig. 8.6), we identified an essential component of 10 states. According to the associated stationary distribution, the most probable states are (SURPRISED, WONDERING) ($p = 0.6286$), (WONDERING, ASTONISHED) ($p = 0.2292$), and (ASTONISHED, DISLIKING) ($p = 0.0917$). As in the previous case, observed frequencies are close to those provided by the stationary distribution.

Similar remarks can be made about *System3*, for which an instance of interaction is shown in Fig. 8.9. The only essential component in this system consists of 19 states, and its stationary distribution identifies, as most probable states, (ANGRY, ANGRY), with $p = 0.2930$, (WORRIED, ANNOYED), with $p = 0.1725$, and (ANNOYED, DISLIKING), with $p = 0.1250$. Taken altogether, these data confirm that the learning process was successful in having A^2 acquire an effective policy, since the goal states defined for A^1 are among the most probable states of the system, in each of the considered examples.

A natural question now is to establish how precisely these stationary distributions describe the actual behavior of the systems. To this purpose, we used property 5 in Section 8.6.1 to compute the minimum number of steps required to approximate the stationary distribution with an arbitrary small error ε. For $\varepsilon = 0.001$, we computed this value for the three examples considered above, and found, respectively, $n_1 = 38.12$, $n_2 = 26.62$, and $n_3 = 42.04$. This means that the stationary distribution is a suitable descriptor of the actual behavior of the above systems even after a limited amount of steps. This also explains why the probability values of the stationary distributions are rather close to the frequencies observed in the experiments.

Properties 3 and 4 of Section 8.6.1 can be used to compute mean entrance times into a given state of interest j starting from another state of interest i, for any i, j. To this purpose, one can define a set of starting states, SS, and a set of end states, ES, and compute the minimum, maximum, and average waiting time to go from SS to ES. More formally, let us call $S_{red} \subseteq S \times S$ the set of states in the essential component we are studying, and let us define $SS \subseteq S_{red}$, $ES \subseteq S_{red}$. Then we can build a matrix of mean entrance times:

$$MET_{SS,\,ES} = \{E_{(a,b)}(\tau_{(c,d)}) \mid (a,b) \in SS, (c,d) \in ES\} \tag{8.18}$$

Depending on how we define SS and ES, we can study the mean waiting time for the system to go, for instance, from (JOYFUL, SAD) to (SAD, JOYFUL). Alternatively, we can focus on a single agent's states, to check, for instance, how many steps are required, on average, for A^1 to go from the JOYFUL state to the SAD one; the same holds for agent A^2.

At first, we applied the analysis of mean entrance times to the simplified interaction scenario we introduced earlier in this section, where a random agent interacts with a friendly partner and no adaptation occurs. We set $SS = \{(a,b) \mid b = \{\text{ANGRY}\},\ a \in S\}$, and $ES = \{(a,b) \mid b = \{\text{JOYFUL}\},\ a \in S\}$. In other words, we were interested in studying how many steps are required for the friendly agent A^2 to go from the ANGRY state to the JOYFUL one, regardless of A^1's state. We computed $MET_{SS,\,ES}$, and found a minimum value of 9.27 and a maximum value of 374.43 (mean 191.53). Therefore, whereas in the best case the JOYFUL state is reached quite quickly, in the worst case scenario reverting the emotional state of A^2, from a negative to a positive one, can take very long. This is due both to the random nature of agent A^1 and to the absence of a strategy in A^1's attitude aimed at making A^2 JOYFUL.

When considering a reinforcement learning scenario, where A^2 interacts with A^1 by adopting the policy it learned for driving A^1 towards some given goal states, it is natural to ask how many interaction steps, on average, will be required for leading A^1 to a goal state, regardless of A^2's states: $ES = \{(a,b) \mid a \text{ is a goal state}, b \in S\}$. The computation of mean entrance times, in this case, provides a measure of effectiveness of the learning process, in terms of how quickly a goal is reached. Since, by default, all our interactions were started from the initial state (NEUTRAL, NEUTRAL), a natural choice for SS would be $SS = \{(\text{NEUTRAL}, b) \mid b \in S\}$.

We, therefore, analyzed mean entrance times for the three systems previously considered, starting with *System1*. Since, in this example, A^2 learned a policy for making A^1 angry most of the time, the set of ending states was defined as $ES = \{(a,b) \mid a = \{\text{ANNOYED, ANGRY, FURIOUS}\},\ b \in S\}$. Note that in this example no state of the form $(\text{NEUTRAL}, b)$ belongs to the essential component, and hence we could not use it as starting state. A natural choice of starting state in this case is (MELANCHOLIC, CONTEMPTUOUS), which seems to be rather far away from the states in ES. Under these assumptions, we computed $MET_{SS,\,ES}$ and found a minimum of 5.91 and a maximum of 213.10 steps, on average, for going from states in SS to states in ES (mean 77.98).

In *System2*, A^2's aim was to make A^1 surprised. Therefore we defined $ES = \{(a,b) \mid a = \{\text{WONDERING, SURPRISED, ASTONISHED}\},\ b \in S\}$. Since the (NEUTRAL, ANGRY) state belongs

to the essential component, we could choose it as the unique starting state. The computation of $MET_{SS,\,ES}$ yielded a minimum of 3.86 and a maximum of 12.43 steps (mean 7.07).

Lastly, we applied the same analysis to *System3*, where goal states are the same as *System1*, and therefore: $ES = \{(a, b) \mid a = \{\text{ANNOYED}, \text{ANGRY}, \text{FURIOUS}\},\ b \in S\}$, $SS = \{(\text{NEUTRAL}, b) \mid b \in S\}$. Given $MET_{SS,\,ES}$, the minimum and the maximum values are 6.58 and 18.14, respectively (mean 11.02).

Therefore, we can conclude that, in the last two examples, goal states are reached very quickly after interaction starts (within $15 - 20$ steps, approximately), which confirms that the learned policies are effective in driving A^1's behavior to the given goals. On the contrary, in order to reach the goal states *System1* seems to require rather a long time (about 78 steps, on average) with respect to the size of the essential component (15, in this case). However, we observe that this is mainly due to two particular end states ((FURIOUS, FURIOUS) and (FURIOUS, NEUTRAL)) that in general have very low entrance probabilities; we note that the other three goal states in this example can be reached within 30 steps, confirming in any case a good performance of the system.

To summarize our results, we used classical properties in Markov chains theory to extract some statistical information about given interaction scenarios. Through the computation of the stationary distribution of the essential component of the system, we extracted the most probable states. These largely correspond to goal states as defined in the reinforcement learning framework. The accuracy of the approximation of the actual distribution on the system to the stationary distribution was computed, too, and found to be rather good since the earliest interaction steps. Mean waiting times for going from state i to state j were used to establish the number of steps required, on average, for the system to reach a set of states of interest. This analysis showed that goal states, as defined in the reinforcement learning framework, are reached rather quickly after interaction starts, confirming that the learned policies are effective.

8.7 Discussion

The model is characterized by three key elements: *personality*, *attitude* and *nature*. These terms were chosen to fit the corresponding psychological elements. *Personality* (Ryckman, 2003) is related to the psychological structure of an individual; it is believed to be mostly stable and independent from external events. In our model, the agent's personality is represented as the transition probabilities matrix given at time zero, P_0. Personality defines therefore the initial interaction behavior of the agent (which emotional states will be assumed, and how frequently) and acts as a basis upon which the agent's attitude can develop. *Attitude* (Zanna and Rempel, 1988) is more related to specific situations and targets (other agents, for instance), shaping the individual's behavior in response to external stimuli. In the presented model, starting from an initial behavior, exclusively defined by the agent's personality, the agent gradually has its behavior changed according to the interaction history, so that the

current emotional interaction behavior results from a combination of personality and a sequence of attitude changes. The current attitude towards the interlocutor is therefore defined as the transition probabilities matrix at time t, P_t. Lastly, we defined *nature* as a driving force used for updating attitude according to the past interaction sequence, to favor a subset of the emotional states, which will be entered more frequently in subsequent interaction steps. We considered, for instance, an imitative and a compensatory nature, driving attitude updates towards two opposite directions.

The resulting model allows carrying out implicitly what is called "affective reasoning" (e.g. André et al., 1999), in which "on the basis of the domain knowledge [...] the appropriate emotional reaction is determined" (Schroeder, 2004). In classical approaches this is achieved by giving to the agent an explicit knowledge of the behavior of the other agent (Ortony et al., 1988). Here, instead, it is the agent itself that discovers the best emotional reactions to the interlocutor, without building any explicit knowledge of it.

In fact, using reinforcement learning it is possible to generate policies for eliciting specific behaviors: the interacting agent will frequently be happy, angry or sad, depending on the goals we have set for the learning phase. Thus, the learning agent will then display ad hoc emotional states for causing its partner to enter desired states. Of course, assuming an emotional state is generally not a matter of decision: it is a spontaneous, often uncontrollable event. Emotional expression, though, can generally be controlled at some level and used to influence the interlocutor. In this sense we believe that reinforcement learning can be applied to emotional interactions in order to define strategies for driving them to desired results. By appropriately setting the interacting personalities, natures and goals, the interaction can be directed toward a general trend, without losing the unpredictable traits of a real emotional interaction.

Emotional states change with continuity and can be viewed as points in a multi-dimensional continuous space (Schlosberg, 1954), which is organized along affective dimensions (such as positive/negative). Emotion gradations can be well captured, as any level of emotion can be represented by a point in this space. In our case, instead, emotional states are discretized into a finite number, and for each state a few discrete levels are considered. For instance, the surprise emotion is represented by three different states (WONDERING, SURPRISED, and ASTONISHED) corresponding to increasing levels of this emotion. Thus, the model cannot entirely account for all different emotion gradations. However, the interaction framework does not require to model a full range of gradations, since even during human-human interaction the displayed emotional states are not perceived at the highest precision, but rather clustered to wider categories (e.g. slight as opposed to intense surprise). While this applies to emotional interaction modeling, fuzzy approaches would probably help when considering more complex processes involving a fine representation of emotional gradations.

It is often postulated (Scherer, 1984b) that emotional states are subject to continuous changes as cognitive appraisal of external stimuli is carried out by the organism. In our model this process has been discretized so that a single emotional state is entered following the

evaluation of the interlocutor's emotion and maintained until the next interaction step, when a new emotional input is fed into the model. We can regard this single emotional state, for each interaction step, as being the only one observable by the other agent as the final result of a continuous emotional process, consisting of intermediate steps that, however, are concealed. Since we are not focusing on reproducing the whole process of emotion production, but rather on modeling emotional interaction, we believe that this discretized approach may be suitable for our aims.

The probabilistic model presented here can be considered as an evolution of previously proposed models. In (Chittaro and Serra, 2004), the term personality has a similar meaning to ours; however, emotional states are not taken into account and no updating of the transition probabilities is included. The model in (Kopecek, 2003) includes inputs and outputs (not just external communications of the emotional state) and no updates. Principally, while our model has a strong time-varying imprint, those frameworks are mainly static. The dependence of transition probabilities on past inputs is considered in (Kuhnlenz and Buss, 2004) through the implementation of a digital (forgetting) filter. However, modifications induced by inputs are not permanent: the impact of a stimulus is effective only during the time interval corresponding to the length of the filter response. Therefore this mechanism does not allow for long-term adaptation of the transition function. In the present work, instead, the transition function for an agent does change in order to adapt to the interaction partner. This reflects an interaction scenario where emotional input from the partner has a strong impact on the agent's attitude, actually shaping it. In order to filter out input variability, time filtering is introduced through the trace mechanism in Eq. 8.1. Our model differs also from that in (Kuhnlenz and Buss, 2004) for other architectural details: our concepts of personality and attitude can be seen as unifying the two matrices of the HMM in a simpler structure; moreover, in our model emotional states are not hidden but directly output to the outside.

In (Nomura, 1996) two of our keywords, attitude and personality, are used, but in different roles. In our model, personality (at start) and attitude select the next emotional state according to the current state and to the input, while in (Nomura, 1996) personality determines the probability of some output given the current state and attitude provides the probability of each next state given current input and output. In our model, this information is merged into one transition probabilities matrix, P_t, representing the current attitude of the agent built over time, starting from the basis defined by its personality. In both models we have attitude updates, but while in (Nomura, 1996) these updates are based on the emotional state and the personality of the individual, in our model attitude is changed according to the interaction history and the agent's nature. Moreover, in our work inputs and outputs are not actions as in (Nomura, 1996), but bodily expressions of the current emotional state. Lastly, the intended aims are quite different. In (Nomura, 1996) the goal is to study the dynamics of the relationships within a group, registering attitude changes over time: that is, to analyse group dynamics. In our work, instead, the aim is to synthesize and to predict emotional behavior in the context of interaction between a human and a robot (or between two generic agents); our

model can also be employed to explore likely interactions between two individuals.

In contrast to the previously discussed works, which lack a robotic implementation, the basic interaction model was implemented on an AIBO robot and therefore experimented also in a real human-robot scenario, thus showing its effectiveness in supporting an emotion-based interaction. Nevertheless, to be applied in more complex human-robot interactions, an accurate detection of human expressions has to be carried out. This is still far beyond reach to actual computer systems as facial movements are small and very fast. In order to have a successful emotional interaction, facial expressions have still to be somehow exaggerated as shown in Fig. 8.2.

8.8 Conclusion

The proposed model allows, thanks to its probabilistic and dynamic nature, to model a wide variety of behaviors occurring during emotional interactions. By adopting the reinforcement learning framework, the model is also able to automatically discover behavioral patterns which adapt to the interlocutor, in order to successfully interact with another agent, without needing any a priori knowledge of it.

The described interaction model has a basic structure that can easily be extended and personalized, by adding or modifying emotional states, inputs, personalities and natures. The model can be used as a basis for emotional agents (e.g. in video games) or robots, in an effort to have technology adapt to its user's characteristics. We can imagine a video game where the user's avatar has to interact with different synthetic agents in order to walk successfully through the game. Such an interaction may be based also on emotional cues, with synthetic agents reacting differently to different emotional inputs, and thus leading to different outcomes for the game itself. In social robotic applications, the robot's personality might be carefully designed to best meet the needs, or simply the preferences, of the user; similarly, starting from a basic personality, user-robot interactions would autonomously shape the robot's attitude according to the user's wishes. The capability of the model of adapting to the input trend – where adaptation can be meant in an imitative, or opposite direction, or in learning how to drive the interlocutor's behavior toward desired goals, or also defined anew by the user – improves the interaction quality, providing lifelike features to its dynamics.

On the other hand, the model allows for behavioral dynamics analyses: high probability cycles can be identified as characteristic patterns for the considered emotional context. Moreover, Markov chains theory can be applied to specific instances of interaction for extracting statistical information about the expected overall behavior of the system, for instance for predicting how frequently, or after how many interaction steps, an emotional state will be entered. Our study is here based on the properties of homogeneous Markov chains, where transition matrices do not change with time. It would be interesting to develop a similar analysis by using nonhomogeneous chains, where transition probabilities change as time goes by.

Concluding Remarks

"Either write something worth reading, or do something worth writing."
— Benjamin Franklin, 1706–1790

Scope and motivations

In this Thesis, three research lines have been presented, namely:

I. analysis of neuroimaging data on the cognitive domain of single word reading;

II. modelling of reading processes in humans; and,

III. simulation of emotional intelligence in artificial systems.

All these topics revolve around a common fulcrum: that is, human intelligence. However, they look at this common ground from different perspectives. Part I and II have focused on the attempt of furthering our understanding of human cognition, and have done so at two different levels of analysis: the meta-analytic approach described in Part I considers the hemodynamic response of small populations of neuronal cells when subjects are engaged in reading tasks (low-level approach); the modelling work builds on higher-level assumptions on the cognitive operations involved in reading (high-level approach). Also, the investigation methods of Part I can be regarded as being data-driven, as they derive interpretations starting from an exploratory analysis of the available data (from data to hypotheses); conversely, the modelling approach described in Part II is hypothesis-driven in that it establishes some basic assumptions on the process to be modelled and implements them in a working simulation; simulated data are then compared to the real ones, so that the validity of the modelled assumptions can be assessed (from hypotheses to data). Part III, on the other hand, represents an attempt to reproduce a specific aspect of human intelligence (i.e. emotional intelligence) in an artificial system; here, human intelligence plays the role of inspiration, of ultimate goal for the designer of an intelligent machine. The model of human intelligence provides a list of *desiderata*, and suggestions on how to achieve them, that guide the building of the system. The choice of three distinct research lines is functional in showing how different paths can be followed in approaching the same ideal destination.

The works presented in this Thesis are also strongly characterized by their commitment to an interdisciplinary approach – combining tools and methods from computer science and neuroscientific-psychological competences – in both reverse-engineering (from behavior to

underlying functional principles) and re-engineering (from functional principles to behavior) human intelligence. Arguably, none of the results included in this Thesis would have been attained without heavily relying on the synergy between these disciplines.

Whereas the first two parts of this Thesis are concerned with the faculty of reading (and therefore of language, at large), the last part considers affective functions. Although it might appear that a huge gap exists between these two topics, it is worth noting that both are high-level faculties that are typical of humans: written language is surely a prerogative of the human kind, and emotional intelligence, even though forms of affective communication exist in other animals too, reaches its highest point in human societies. Both language and affective behavior, thus, strongly characterize human intelligence. The choice of these particular topics was therefore motivated by the desire to understand and reproduce those faculties that make the human mind such a unique system.

Major contributions

Each part of this Thesis was organized to couple an extensive review of the literature on the targeted topic with a set of novel results. Particular care was given to the review sections, in an attempt to provide, as far as possible, a self-contained treatment of each topic; we figured that this could be particularly useful to the "less-interdisciplinary" reader, who might find in these surveys a set of handy pointers to learn more about a particular subfield. Each part is also characterized by its own major contribution.

In Part I, we have presented two main results. On the methodological side, we have developed a new hierarchical clustering algorithm that improves on standard implementations, as it successfully addresses the problem of non-uniqueness of the solution due to the presence of ties in the dissimilarity matrix. By analyzing tie configurations and defining an equivalence relation over dendrograms, the significantly different solutions associated with a dataset are generated; then, these solutions are compared to find the best one according to a user-defined measure. In this way, a unique solution, up to equivalences, is always returned for each dataset. This allows for more controlled analyses in clustering-based studies (e.g. in bioinformatics), where different clustering solutions might be associated with different interpretations of the same data. On the experimental side, we have described a meta-analysis, employing this algorithm, of neuroimaging data on single word reading published in the scientific literature from 1992 to 2008. The results from this work contribute to forming a clearer picture of the organization of the neural circuitry underlying reading processes. From this meta-analysis the picture of a widely distributed network emerges where no strict anatomical segregation for lexical and sub-lexical processes could be found; this finding contrasts with the popular dual-route theory of reading, whereas it provides some support for the alternative, single-mechanism account, on which the connectionist "triangle" models are based.

Connectionist modelling is, in fact, an effective approach to investigate a cognitive function. In Part II of this Thesis we have taken this approach in our investigation of reading processes: our contribution lies in the design, implementation, and testing of a new connec-

tionist model, called the two-component model. This model assumes that information flows, in a cascaded fashion, from a cognitive component to an articulatory one; while the former is devoted to computing a distributed internal representation for the input word, the latter is responsible for turning the internal code into the corresponding sequence of phonemes (the pronunciation of the word). The code computed by the cognitive component gets clearer over time and, when strong enough, it triggers the initiation of the response by the articulatory component. We have proposed that behavioral effects on reaction times in experimental subjects (especially the so-called serial effects) might be the result of such a mechanism. As the model is still in its infancy, simulation results are not conclusive yet, and call for additional developments. Here, we have striven to provide a critical and detailed analysis of the strengths and weaknesses of the proposed implementation, and indicate potentially effective ways to improve it.

Finally, the major contribution in Part III consists of a new model for simulating emotional interaction between a human user and a robot, or between two synthetic agents. In this work, agents interact by exchanging emotional states: for each artificial agent, the current emotional state is generated by a PFSA based on the previous state and the current input from the interacting partner. Transition probabilities encode the personality for the agent, so that different affective behaviors are associated with different personalities. These probabilities, moreover, change over time in response to the history of the interaction (in particular, we have shown how this can be attained by reinforcement learning), and this allows for an autonomous adaptation of an agent's affective behavior to the one displayed by its interacting partner. This feature, together with the non-deterministic mechanism for generating emotional states, make the resulting interactions more lifelike and natural: as emotional responses are not hard-coded and immutable (which might lead the human user to perceive them as markedly artificial), they are less predictable and, therefore, closer to human reactions. This model provides a way to improve user-machine interfaces by endowing them with affective competences, which enable enhanced interactive experiences: arguably, the possibility of exploiting an emotional channel for communication, in addition to more standard channels, holds the potentiality for more user-friendly and effective user-machine interaction. A relevant contribution lies also in the analytic tools, based on Markov chain theory, that we have provided for extracting quantitative descriptions of the interaction scenarios generated by our model. From a more general perspective, this work represents a contribution to the advancement of AI: as emotional intelligence is indeed an important part of human intellective faculties, the introduction of enhanced affective competences in artificial systems should, in the long range, help achieve more realistic and powerful artificial intelligence.

Future work

In closing, we summarize some directions for future developments of the work presented in this Thesis. The methodological work on meta-analysis of neuroimaging data described in Part I could benefit from careful code optimization, so as to reduce execution times and make

the clustering algorithm more efficient also for very large datasets. Work is already under way for developing an integrated suite for meta-analysis to be released to the neuroimaging community, so that all phases of the analysis (input acquisition, data normalization, clustering, anatomical labelling, and visualization) can be managed in a simple, semi-automatic way. Hopefully the release of this tool will encourage other research groups to perform new meta-analyses, and thus contribute to advancing our knowledge of the functional organization of the brain. There is space for improvements in the anatomical segregation module, as well: for instance, more sophisticated approaches could be devised that can handle weak constraints (recommendations, rather than absolute forbiddances) on boundary-crossing clusters.

In Part II we have discussed several ways to extend and improve our connectionist model of reading. These include a complete implementation of the orthographic input section, retraining of the model for a limited number of epochs (so as to control for overtraining), and possibly the employment of other quality measures for assessing the clarity of the internal code computed by the network. This additional work would result in a more mature and stable model, whose plausibility in the face of the behavioral data reported in the reading literature could be tested in a more definite way.

Further investigations on the emotional interaction model proposed in Part III can be delineated too. The model could be extended by explicitly handling the external manifestations of the experienced emotional state. Let us consider the two-synthetic-agent setting: in the current formulation each agent directly observes the emotional state of the interacting partner; in other words, an agent's output is simply an emotional label. However, affective states are hardly observed in such an open way, and they rather have to be inferred on the basis of behavior (e.g. facial expressions, gestures) and physiological modifications (e.g. becoming pale); moreover, as people sometimes try to suppress behaviors that might reveal their inner state, or even act to pretend a different state than the actual one, the correspondence between an emotional label and manifestations associated with it should not be one-to-one. Rather, we might argue that the way these correspondences are defined should depend on the agent's personality, so that one particular agent might be more transparent than a different one. This would enrich the model and increase its ability to produce credible interactive behaviors. Finally, it would be interesting to explicitly test to what degree the interaction scenarios produced by our model are perceived as plausible by human observers: experimental subjects could be asked to interact with two agents, one based on our model, and the other having hard-coded, pre-programmed affective responses (other agents employing different models proposed in the literature might also be included in the experiment). If subjects rated the interactions they had with our model to be consistently more convincing and humanlike, our model could be claimed to be successful in producing natural interactive experiences.

Appendix A

"Begin at the beginning," the King said gravely, "and go on till you come to the end: then stop."

— From "Alice in Wonderland", by Lewis Carroll, 1832–1898

A.1 Ward's dissimilarity measure and the increase in the error sum of squares

In this section we will provide an explicit proof of the fact that in hierarchical clustering the choice of the pair of clusters having minimum Ward's dissimilarity (Ward, 1963) is equivalent to minimizing the increase in the error sum of squares (ΔESS, see Eqs. 2.11 and 2.12) for the clustering solution at the current level of the hierarchy, due to the current merging operation. This holds true if

$$diss(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2 ,$$

that is, if the dissimilarity measure for pairs of single data points is the squared Euclidean distance. In this case:

$$Diss_C(C_i, C_j) = 2\Delta ESS_{i,j}$$

for any two clusters C_i, C_j, with $Diss_C$ denoting here Ward's dissimilarity measure.

Proof. For the sake of simplicity we will work in one dimension (so x, y, and z are not feature vectors, but points in the two-dimensional space); we start by re-writing ESS_i in another form, that is

$$ESS_i = \sum_{x \in C_i} x^2 - \frac{1}{n_i} \left(\sum_{x \in C_i} x \right)^2$$

where n_i is, as usual, the cardinality of cluster C_i. Let us now consider a generic clustering step where C_1 and C_2 have been merged to form cluster C_3, and we are interested in knowing how much the error sum of squares of our clustering solution would increase if we were to merge C_3 to another cluster, C_o (o for denoting a generic other cluster):

$$\Delta ESS_{3,o} = ESS_{3,o} - ESS_3 - ESS_o$$

Since $C_3 = C_1 \cup C_2$, $ESS_{3,o} = ESS_{1,2,o}$, and $ESS_3 = ESS_{1,2}$. Therefore we can re-write:

$$\Delta ESS_{1,2,o} = ESS_{1,2,o} - ESS_{1,2} - ESS_o = \sum_{x\in C_1} x^2 + \sum_{y\in C_2} y^2 + \sum_{z\in C_o} z^2 - \frac{1}{n_1+n_2+n_o}\left(\sum_{x\in C_1} x + \sum_{y\in C_2} y + \sum_{z\in C_o} z\right)^2 +$$
$$- \left(\sum_{x\in C_1} x^2 + \sum_{y\in C_2} y^2 - \frac{1}{n_1+n_2}\left(\sum_{x\in C_1} x + \sum_{y\in C_2} y\right)^2\right) - \left(\sum_{z\in C_o} z^2 - \frac{1}{n_o}\left(\sum_{z\in C_o} z\right)^2\right) = \frac{1}{(n_1+n_2+n_o)(n_1+n_2)n_o}$$
$$\left(-(n_1+n_2)n_0\left(\left(\sum_{x\in C_1} x\right)^2 + \left(\sum_{y\in C_1} y\right)^2 + \left(\sum_{z\in C_o} z\right)^2 + 2\left(\sum_{x\in C_1} x\right)\left(\sum_{y\in C_2} y\right) + 2\left(\sum_{x\in C_1} x\right)\left(\sum_{z\in C_o} z\right) +\right.\right.$$
$$\left.\left. +2\left(\sum_{y\in C_2} y\right)\left(\sum_{z\in C_o} z\right)\right) + (n_1+n_2+n_o)n_o\left(\left(\sum_{x\in C_1} x\right)^2 + \left(\sum_{y\in C_2} y\right)^2 + 2\left(\sum_{x\in C_1} x\right)\left(\sum_{y\in C_2} y\right)\right) +\right.$$
$$\left. +(n_1+n_2+n_o)(n_1+n_2)\left(\left(\sum_{z\in C_o} z\right)^2\right)\right) = \frac{1}{(n_1+n_2+n_o)(n_1+n_2)n_o}\left((-n_1n_o - n_2n_o + n_1n_o + n_2n_o + n_o^2)\right.$$
$$\left(\sum_{x\in C_1} x\right)^2 + (-n_1n_o - n_2n_o + n_1n_o + n_2n_o + n_o^2)\left(\sum_{y\in C_2} y\right)^2 + (-n_1n_o - n_2n_o + n_1^2 + 2n_1n_2 + n_2^2 + n_1n_o + n_2n_o)$$
$$\left(\sum_{z\in C_2} z\right)^2 + 2(-n_1n_o - n_2n_o + n_1n_o + n_2n_o + n_o^2)\left(\sum_{x\in C_1} x\right)\left(\sum_{y\in C_2} y\right) + 2(-n_1n_o - n_2n_o)\left(\sum_{x\in C_1} x\right)\left(\sum_{z\in C_o} z\right) +$$
$$\left. +2(-n_1n_o - n_2n_o)\left(\sum_{y\in C_2} y\right)\left(\sum_{z\in C_o} z\right)\right) = \frac{1}{(n_1+n_2+n_o)(n_1+n_2)n_o}\left(n_o^2\left(\sum_{x\in C_1} x + \sum_{y\in C_2} y\right)^2 + (n_1+n_2)^2\right.$$
$$\left.\left(\sum_{z\in C_o} z\right)^2 + 2(-n_1n_o - n_2n_o)\left(\sum_{z\in C_o} z\right)\left(\sum_{x\in C_1} x + \sum_{y\in C_2} y\right)\right)$$

Thus,

$$\Delta ESS_{3,o} = \frac{1}{(n_1+n_2+n_o)}\left(\frac{n_o}{n_1+n_2}\left(\sum_{x\in C_1} x + \sum_{y\in C_2} y\right)^2 + \frac{n_1+n_2}{n_o}\left(\sum_{z\in C_o} z\right)^2 +\right.$$
$$\left. - 2\left(\sum_{z\in C_o} z\right)\left(\sum_{x\in C_1} x + \sum_{y\in C_2} y\right)\right) \tag{A.1}$$

After these preliminary steps, we can now prove that $Diss_C(C_3, C_o) = 2\Delta ESS3, o$. Using induction, we can start by considering the base case, where both C_3 and C_o are singletons (they contain one element only, respectively a and b). In this case, by definition $Diss_C(C_3, C_o) = diss(a, b)$:

$$Diss_C(C_3, C_o) = (a-b)^2 = a^2 + b^2 - 2ab$$
$$\Delta ESS_{3,o} = ESS_{3,o} = \left(a - \frac{a+b}{2}\right)^2 + \left(b - \frac{a+b}{2}\right)^2 = \frac{(a-b)^2}{2} = \frac{1}{2}(a^2 + b^2 - 2ab)$$

Now, for the inductive step let us assume that each of the dissimilarity values required by the Lance-Williams formula for Ward's method (see Eqs. 2.13 and 2.23) are actually equal to the corresponding ΔESS value (note that we drop here the multiplicative constant 2 for simplicity, but without any loss

of generality), so, for instance

$$Diss_C(C_1, C_o) = \Delta ESS_{1,o} = ESS_{1,o} - ESS_1 - ESS_o = \\ = \sum_{x \in C_1} x^2 + \sum_{z \in C_o} z^2 - \frac{1}{n_1 + n_o} \left(\sum_{x \in C_1} x + \sum_{z \in C_o} z \right)^2 + \\ - \sum_{x \in C_1} x^2 + \frac{1}{n_1} \left(\sum_{x \in C_1} x \right)^2 - \sum_{z \in C_o} z^2 + \frac{1}{n_o} \left(\sum_{z \in C_o} z \right)^2$$

Assuming this holds, we can consider the next clustering step (the one involving the computation of $Diss_C(C_3, C_o)$) to prove that computing this value yields Eq. A.1.

$$Diss_C(C_3, C_o) = \frac{1}{n_1 + n_2 + n_o} \left((n_1 + n_o) \left(-\frac{1}{n_1 + n_o} \left(\sum_{x \in C_1} x + \sum_{z \in C_o} z \right)^2 + \frac{1}{n_1} \left(\sum_{x \in C_1} x \right)^2 + \frac{1}{n_o} \left(\sum_{z \in C_o} z \right)^2 \right) + \right.$$
$$+ (n_2 + n_o) \left(-\frac{1}{n_2 + n_o} \left(\sum_{y \in C_2} y + \sum_{z \in C_o} z \right)^2 + \frac{1}{n_2} \left(\sum_{y \in C_2} y \right)^2 + \frac{1}{n_o} \left(\sum_{z \in C_o} z \right)^2 \right) - (n_o) \left(-\frac{1}{n_1 + n_2} \right.$$
$$\left. \left. \left(\sum_{x \in C_1} x + \sum_{y \in C_2} y \right)^2 + \frac{1}{n_1} \left(\sum_{x \in C_1} x \right)^2 + \frac{1}{n_2} \left(\sum_{y \in C_2} y \right)^2 \right) \right) = \frac{1}{n_1 + n_2 + n_o} \left(- \left(\sum_{x \in C_1} x + \sum_{z \in C_o} z \right)^2 + \right.$$
$$\left. - \left(\sum_{y \in C_2} y + \sum_{z \in C_o} z \right)^2 + \frac{n_o}{n_1 + n_2} \left(\sum_{x \in C_1} x + \sum_{y \in C_2} y \right)^2 + \left(\sum_{x \in C_1} x \right)^2 + \left(\sum_{y \in C_2} y \right)^2 + \frac{n_1 + n_2 + 2n_o}{n_o} \left(\sum_{z \in C_o} z \right)^2 \right) =$$
$$= \frac{1}{n_1 + n_2 + n_o} \left(- \left(\sum_{x \in C_1} x \right)^2 - \left(\sum_{z \in C_o} z \right)^2 - 2 \left(\sum_{x \in C_1} x \right) \left(\sum_{z \in C_o} z \right) - \left(\sum_{y \in C_2} y \right)^2 - \left(\sum_{z \in C_o} z \right)^2 + \right.$$
$$\left. -2 \left(\sum_{y \in C_2} y \right) \left(\sum_{z \in C_o} z \right) + \frac{n_o}{n_1 + n_2} \left(\sum_{x \in C_1} x + \sum_{y \in C_2} y \right)^2 + \left(\sum_{x \in C_1} x \right)^2 + \left(\sum_{y \in C_2} y \right)^2 + \frac{n_1 + n_2 + 2n_o}{n_o} \left(\sum_{z \in C_o} z \right)^2 \right).$$

The last re-writing step bring us to

$$Diss_C(C_3, C_o) = \frac{1}{n_1 + n_2 + n_o} \left(\frac{n_o}{n_1 + n_2} \left(\sum_{x \in C_1} x + \sum_{y \in C_2} y \right)^2 + \frac{n_1 + n_2}{n_o} \left(\sum_{z \in C_o} z \right)^2 + \right. \\ \left. - 2 \left(\sum_{z \in C_o} z \right) \left(\sum_{x \in C_1} x + \sum_{y \in C_2} y \right) \right),$$

that is, Eq. A.1. □

A.2 Applicability of the proposed clustering algorithm to other dissimilarity measures

In this section we will briefly discuss the applicability of our HC method to other linkage methods than Ward's. In order for our method to be applied to a specific HC algorithm, the monotonicity of the sequence of merging coefficients must be assessed; for Ward's method, this has already been proved (see Theorem 2.3.3).

Fact A.2.1. *Single, complete, group average, and weighted group average linkage produce monotonic sequences of merging coefficients; this does not hold for centroid and median linkage.*

Proof. Let $v = Diss_C(C_i, C_j)$ be the minimum value in the current dissimilarity matrix H; C_i and C_j are the two clusters being merged. For any given $Diss_C$, we must prove that for any $C_k \neq C_i, C_j$, $Diss_C(\{C_i, C_j\}, C_k) \geq v$. We will use the observation that, being v the minimum value in H, it follows that for any C_k, $d_{ik} = Diss_C(C_i, C_k) \geq v$ and $d_{jk} = Diss_C(C_j, C_k) \geq v$.

- Single linkage:

  $$Diss_C(\{C_i, C_j\}, C_k) = \min\{d_{ik}, d_{jk}\}.$$

  Monotonicity follows from the above observation.

- Complete linkage:

  $$Diss_C(\{C_i, C_j\}, C_k) = \max\{d_{ik}, d_{jk}\}.$$

  As above.

- Weighted group average linkage:

  $$Diss_C(\{C_i, C_j\}, C_k) = \frac{1}{2}d_{ik} + \frac{1}{2}d_{jk}.$$

  As above.

- Group average linkage:

  $$Diss_C(\{C_i, C_j\}, C_k) = \frac{n_i}{n_i + n_j}d_{ik} + \frac{n_j}{n_i + n_j}d_{jk} \geq \frac{n_i v + n_j v}{n_i + n_j} = v.$$

- Centroid linkage:

  $$\begin{aligned} &Diss_C(\{C_i, C_j\}, C_k) = \frac{n_i}{n_i + n_j}d_{ik} + \frac{n_j}{n_i + n_j}d_{jk} - \frac{n_i n_j}{(n_i + n_j)^2}v = \\ &= \frac{(n_i^2 + n_i n_j)d_{ik} + (n_j^2 + n_i n_j)d_{jk} - n_i n_j v}{(n_i + n_j)^2} = \frac{n_i^2 d_{ik} + n_j^2 d_{jk} + n_i n_j (d_{ik} + d_{jk} - v)}{(n_i + n_j)^2} = \\ &= \frac{n_i^2(v + \epsilon) + n_j^2(v + \eta) + n_i n_j (v + \epsilon + \eta)}{(n_i + n_j)^2} = \\ &= \frac{(n_i + n_j)^2 v - n_i n_j v + (n_i^2 + n_i n_j)\epsilon + (n_j^2 + n_i n_j)\eta}{(n_i + n_j)^2} = v + \frac{n_i^2 \epsilon + n_j^2 \eta + n_i n_j (\epsilon + \eta - v)}{(n_i + n_j)^2}. \end{aligned}$$

  Therefore, if $v > \left(\frac{n_i}{n_j} + 1\right)\epsilon + \left(\frac{n_j}{n_i} + 1\right)\eta$, then $Diss_C(\{C_i, C_j\}, C_k) < v$.

- Median linkage:

  $$Diss_C(\{C_i, C_j\}, C_k) = \frac{1}{2}d_{ik} + \frac{1}{2}d_{jk} - \frac{1}{4}v = \frac{1}{2}(v + \epsilon) + \frac{1}{2}(v + \eta) - \frac{1}{4}v = \frac{3}{4}v + \frac{1}{2}(\epsilon + \eta).$$

  If $v > 2(\epsilon + \eta)$, then $Diss_C(\{C_i, C_j\}, C_k) < v$.

Note that, if C_i, C_j is a non-critical pair, both d_{ik} and d_{jk} must be $> v$. It follows that, for single, complete, group average, and weighted group average linkage, $Diss_C(\{C_i, C_j\}, C_k) > v$. □

Let us also note that property 2.24c, that is basically an *associativity* property, in Ward's method always holds true for critical pairs.

Fact A.2.2. *Equation 2.24c always holds for critical pairs C_i, C_j and C_j, C_k if single, complete, weighted group average, or median linkage is employed; this does not hold for group average and centroid linkage.*

Proof. Let us recall that, by definition of critical pairs, $v = Diss_C(C_i, C_j) = Diss_C(C_j, C_k)$. For any given $Diss_C$ we wish to prove whether $Diss_C(\{C_i, C_j\}, C_k) = Diss_C(\{C_j, C_k\}, C_i)$ always holds true. For brevity reasons, let us write $d_{ik} = Diss_C(C_i, C_k)$.

- Single linkage:

$$Diss_C(\{C_i, C_j\}, C_k) = \frac{1}{2}d_{ik} + \frac{1}{2}v - \frac{1}{2}\left|d_{ik} - v\right|.$$

  Since $d_{ik} \geq v$ (v is the minimum dissimilarity value), this can be rewritten as

$$\frac{1}{2}d_{ik} + \frac{1}{2}v - \frac{1}{2}d_{ik} + \frac{1}{2}v = v.$$

  As for the other side of the equality:

$$Diss_C(\{C_j, C_k\}, C_i) = \frac{1}{2}v + \frac{1}{2}d_{ik} - \frac{1}{2}\left|d_{ik} - v\right| = v.$$

- Complete linkage:

$$Diss_C(\{C_i, C_j\}, C_k) = \frac{1}{2}d_{ik} + \frac{1}{2}v + \frac{1}{2}\left|d_{ik} - v\right| = d_{ik} = Diss_C(\{C_j, C_k\}, C_i).$$

- Weighted group average linkage:

$$Diss_C(\{C_i, C_j\}, C_k) = \frac{1}{2}d_{ik} + \frac{1}{2}v = Diss_C(\{C_j, C_k\}, C_i).$$

- Group average linkage:

$$Diss_C(\{C_i, C_j\}, C_k) = \frac{n_i}{n_i + n_j}d_{ik} + \frac{n_j}{n_i + n_j}v = \frac{n_i d_{ik} + n_j v}{n_i + n_j}.$$

  On the other side of the equality:

$$Diss_C(\{C_j, C_k\}, C_i) = \frac{n_j}{n_j + n_k}v + \frac{n_k}{n_j + n_k}d_{ik} = \frac{n_j v + n_k d_{ik}}{n_j + n_k}.$$

  The equality holds true only when $n_i = n_k$ or $d_{ik} = v$.

- Centroid linkage:

$$Diss_C(\{C_i, C_j\}, C_k) = \frac{n_i}{n_i + n_j}d_{ik} + \frac{n_j}{n_i + n_j}v - \frac{n_i n_j}{(n_i + n_j)^2}v = \frac{(n_i^2 + n_i n_j)d_{ik} + n_j^2 v}{(n_i + n_j)^2}.$$

On the other side of the equality:

$$Diss_C(\{C_j, C_k\}, C_i) = \frac{n_j}{n_j + n_k} v + \frac{n_k}{n_j + n_k} d_{ik} - \frac{n_j n_k}{(n_j + n_k)^2} v = \frac{n_j^2 v + (n_k^2 + n_j n_k) d_{ik}}{(n_j + n_k)^2}.$$

The equality holds true if $n_i = n_k$, but it does not in the general case.

- Median linkage:

$$Diss_C(\{C_i, C_j\}, C_k) = \frac{1}{2} d_{ik} + \frac{1}{2} v - \frac{1}{4} v = \frac{1}{2} d_{ik} + \frac{1}{2} v = Diss_C(\{C_j, C_k\}, C_i).$$

□

It follows that for single, complete, and weighted group average linkage (median linkage is not considered here because it fails to satisfy the monotonicity requirement), the definition of equivalent pairs can be limited to (2.24a) and (2.24b) only, as (2.24c) directly follows. Instead, for (unweighted) group average linkage, property 2.24c must be separately assessed: if this does not hold, then the considered pairs are non-equivalent (even if properties 2.24a and 2.24b hold).

As a closing note, we observe that the least sensitive method to multiple solutions is single linkage, as all critical pairs are equivalent (according to Def. 2.3.4). Therefore, our algorithm would find one equivalence class and thus produce a single dendrogram in this case. In fact, if (C_i, C_j) and (C_j, C_k) are two critical pairs at step t, this means that C_j includes two objects c_1, c_2 such that $v = Diss_C(c_1, c_i) = Diss_C(c_2, c_k)$, where c_i and c_k are the closest objects to C_j in C_i and C_k respectively, and v is the current minimum value in the dissimilarity matrix. Whichever pair is selected for merging, at $t + 1$ the minimum dissimilarity value is again v, corresponding to the dissimilarity between the new cluster and the one that was not selected at the previous time step.

A.3 Supplementary Materials* for Chapter 3

Non-differentiated clusters

Any cluster that could not be assigned, even in a qualitative way, to one of the psycholinguistic categories we have previously identified (i.e., word-related, pseudoword-related, difficulty-sensitive) was included in the non-differentiated category. These 27 clusters are listed in Table A.3 and shown in Figure A.1.

Among these clusters, we attempted to highlight those having higher probability of actually being completely aspecific. Of course, proving the absence of an effect (the null hypothesis) is more complex than proving its presence: to identify which of these 27 clusters were likely to be significantly non-differentiated, we performed binomial tests along the lexicality axis and for the difficulty effect, yielding three P-values (one for word-sensitivity, one for pseudoword-sensitivity, and one for difficulty-sensitivity) for each cluster. We propose that clusters whose minimum P-value was still greater than 0.4 are of high chance of being genuinely non-differentiated.

Eight clusters survived this strict approach. Four of them are located in the left hemisphere: these are an inferior frontal gyrus (pars orbitalis) cluster, a middle occipital gyrus, and two clusters in the

*Submitted to accompany (Cattinelli et al., Under Revision–a).

Anatomical Label	*MeanX*	*MeanY*	*MeanZ*	*AvgStdX*	*AvgStdY*	*AvgStdZ*	*Num*
Frontal_Sup_L_9	-15	45	36	7	6	6	10
Frontal_Mid_L_6	-29	7	51	7	13	7	26
Frontal_Inf_Orb_L_47	-38	26	-6	7	8	7	37
Frontal_Inf_Tri_L_45	-46	31	14	4	4	8	17
Frontal_Inf_Oper_L	-35	15	16	4	6	8	29
Frontal_Inf_Oper_L_6	-51	9	14	4	7	8	43
Supp_Motor_Area_L_6	0	-1	60	5	5	5	30
Supp_Motor_Area_L_8	-3	26	51	6	6	7	19
Precentral_L_6	-49	-2	31	5	7	5	40
Cingulum_Ant_L_10	-9	48	0	9	6	14	13
Cingulum_Ant_L_11	-6	27	-7	9	7	10	19
Postcentral_L_6	-47	-9	45	4	6	7	29
Parietal_Sup_L_7	-25	-66	51	8	8	6	24
Parietal_Inf_L_40	-49	-40	47	5	10	5	21
Angular_L_40	-35	-49	34	5	5	6	21
Angular_L_39	-44	-75	31	5	8	7	16
Precuneus_L	-3	-57	30	7	4	6	13
Temporal_Sup_L_22	-56	-16	3	6	9	7	29
Temporal_Mid_L_21	-51	-51	17	8	10	7	28
Temporal_Mid_L_21	-57	-45	-1	5	8	6	19
Temporal_Mid_L_20	-61	-18	-21	4	5	5	9
Temporal_Inf_L_20	-48	-46	-19	9	4	4	17
Fusiform_L_37	-28	-39	-12	5	8	6	12
Fusiform_L_19	-36	-80	-12	5	6	2	5
Fusiform_L_20	-33	-25	-25	8	13	8	21
Occipital_Mid_L_19	-23	-73	27	6	6	4	14
Occipital_Mid_L_18	-29	-89	8	6	10	4	18
Occipital_Inf_L_18	-18	-93	-11	10	5	5	19
Occipital_Inf_L_37	-41	-64	-11	6	4	4	26
Lingual_L_18	-17	-73	-7	5	6	3	4
Cerebelum_6_L	-12	-64	-19	5	4	5	8
Cerebelum_6_L	-21	-49	-27	4	8	6	10
Cerebelum_Crus1_L	-31	-80	-23	9	7	4	22
Cerebelum_Crus1_L	-42	-57	-28	5	5	3	8
Thalamus_L	-16	-13	6	10	12	7	31

Table A.1: Left-hemisphere clusters (35). For each cluster, its anatomical label (AAL region and Brodmann area), mean coordinate, average standard deviation, and number of included activation peaks are reported.

Anatomical Label	*MeanX*	*MeanY*	*MeanZ*	*AvgStdX*	*AvgStdY*	*AvgStdZ*	*Num*
Frontal_Mid_R_9	35	12	53	8	7	4	8
Frontal_Mid_Orb_R_47	34	48	-3	6	8	9	6
Frontal_Inf_Tri_R_45	43	30	28	11	11	7	20
Frontal_Inf_Tri_R_45	49	19	5	8	7	8	21
Postcentral_R_4	42	-18	45	7	9	8	23
Postcentral_R_4	50	-3	32	6	10	7	26
Parietal_Inf_R_40	45	-50	43	7	6	6	12
Angular_R_7	30	-62	49	7	8	6	28
Cingulum_Mid_R_24	2	8	41	8	9	8	38
Cingulum_Mid_R_23	3	-38	38	7	5	10	14
Heschl_R	56	-8	8	4	7	7	15
Temporal_Mid_R_22	55	-37	4	5	9	9	29
Fusiform_R_37	39	-53	-16	6	6	5	12
Hippocampus_R_20	35	-7	-16	10	8	8	7
Occipital_Mid_R_39	38	-69	25	10	7	8	15
Occipital_Mid_R_19	36	-82	2	5	5	8	13
Occipital_Inf_R_19	46	-69	-12	6	5	4	10
Calcarine_R_17	6	-96	6	13	7	7	18
Cuneus_R_19	12	-85	30	7	5	9	10
Lingual_R_17	3	-66	10	10	10	5	19
Lingual_R_18	25	-91	-12	3	4	3	11
Cerebelum_6_R	8	-73	-17	7	6	6	17
Cerebelum_6_R	28	-52	-22	6	5	6	12
Cerebelum_Crus1_R	21	-85	-25	6	3	8	11
Cerebelum_Crus1_R	39	-70	-27	5	5	5	7
Vermis_8	5	-64	-35	8	11	9	21
Thalamus_R	18	-21	6	11	10	9	18
Putamen_R	29	14	4	4	7	9	20
Pons/Cerebellum_R	11	-28	-25	14	5	6	8

Table A.2: Right-hemisphere clusters (29).

cerebellum. As for the right hemisphere, we have an inferior parietal cluster, a hippocampus cluster, and two cerebellum clusters.

It is worth noting that the non-differentiated clusters do not necessarily identify regions of conjunctions of activations across stimuli of different psycholinguistic nature; rather, they most likely represent brain regions brought into the dataset because of relatively aspecific control condition used as a baseline for the task of reading during the imaging experiments considered.

Task-specific networks

We also assessed whether a task specific effect could be detected in a given cluster. To this end, we performed a statistical analysis on each cluster using binomial testing, similarly to what was done for assessing lexicality and difficulty effects. In particular we compared tasks according to the following

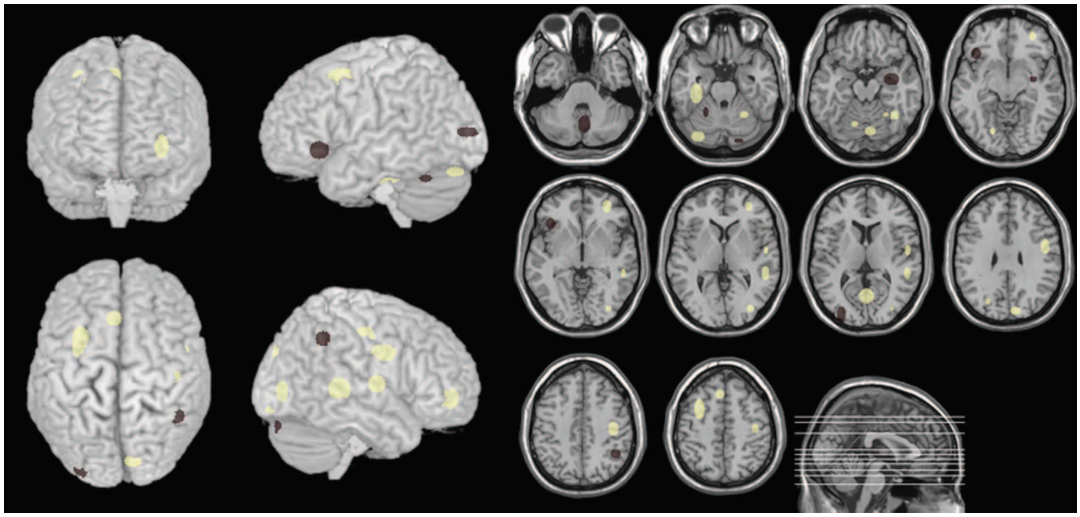


Figure A.1: Clusters in the non-differentiated category. Those clusters that are more likely to be completely non-differentiated in their processing are shown in black.

Anatomical Label	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*	*Anatomical Label*	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*
Frontal_Mid_L_6	-29	7	51	9	26						
						Frontal_Mid_Orb_R_47	34	48	-3	8	6
Frontal_Inf_Orb_L_47*	-38	26	-6	8	37						
Supp_Motor_Area_L_8	-3	26	51	6	19						
						Postcentral_R_4	42	-18	45	8	23
						Postcentral_R_4	50	-3	32	8	26
						Parietal_Inf_R_40*	45	-50	43	6	12
						Heschl_R	56	-8	8	6	15
						Temporal_Mid_R_22	55	-37	4	8	29
						Fusiform_R_37	39	-53	-16	6	12
Fusiform_L_20	-33	-25	-25	10	21						
						Hippocampus_R*	35	-7	-16	9	7
Occipital_Mid_L_19	-23	-73	27	5	14						
Occipital_Mid_L_18*	-29	-89	8	7	18	Occipital_Mid_R_19	36	-82	2	6	13
						Cuneus_R_19	12	-85	30	7	10
						Lingual_R_17	3	-66	10	8	19
Lingual_L_18	-17	-73	-7	5	4	Lingual_R_18	25	-91	-12	3	11
Cerebelum_6_L	-12	-64	-19	5	8	Cerebelum_6_R	8	-73	-17	6	17
Cerebelum_6_L*	-21	-49	-27	6	10	Cerebelum_6_R	28	-52	-22	6	12
Cerebelum_Crus1_L	-31	-80	-23	7	22	**Cerebelum_Crus1_R***	21	-85	-25	6	11
Cerebelum_Crus1_L*	-42	-57	-28	4	8						
						Vermis_8*	5	-64	-35	10	21

Table A.3: Clusters that were not included in any of the previous categories (i.e., difficulty, word, or pseudoword) were considered to be non-differentiated. Among these, we report in bold, and marked by a *, those clusters for which this aspecificity is more likely to be statistically significant.

Anatomical Label	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*	*Anatomical Label*	*MeanX*	*MeanY*	*MeanZ*	*AvgStd*	*Num*
Reading aloud > Silent reading											
						Postcentral_R_4	42	-18	45	8	23
						Heschl_R	56	-8	8	6	15
Thalamus_L_0	-16	-13	6	10	31	Thalamus_R_0	18	-21	6	10	18
Silent reading > Reading aloud											
Frontal_Sup_L_9	-15	45	36	6	10						
Frontal_Inf_Orb_L_47	-38	26	-6	8	37						
Supp_Motor_Area_L_8	-3	26	51	6	19						
Temporal_Mid_L_20	-61	-18	-21	5	9						
Occipital_Mid_L_18	-29	-89	8	7	18						
Reading > Lexical decision											
						Heschl_R	56	-8	8	6	15
Occipital_Mid_L_19	-23	-73	27	5	14						
Cerebelum_Crus1_L	-31	-80	-23	7	22						
Lexical decision > Reading											
Frontal_Mid_L_6	-29	7	51	9	26						
						Frontal_Inf_Tri_R_45	43	30	28	10	20
						Cingulum_Mid_R_23	3	-38	38	7	14
Parietal_Inf_L_40	-49	-40	47	7	21						
Angular_L_39	-44	-75	31	6	16						
Temporal_Mid_L_21	-57	-45	-1	6	19						
Occipital_Mid_L_18	-29	-89	8	7	18						

Table A.4: Clusters showing a reliable effect of task as revealed by binomial tests ($p \leq 0.05$).

classification: reading aloud ($\pi = 0.40$) versus reading silently[1] ($\pi = 0.60$); reading ($\pi = 0.80$) versus lexical decision ($\pi = 0.20$). A task effect was considered to be significant if the computed P-value was less or equal than 0.05.

Table A.4 lists clusters showing a task effect (silent vs. aloud reading, or reading vs. lexical decision). These clusters are also identified by appropriate labelling in the tables that report the psycholinguistically oriented classification of the results.

There were four clusters significantly associated with **reading aloud** rather than reading silently (Figure A.2, blue blobs): both thalami, the right Heschl's gyrus and the right post-central gyrus. Among these, the right Heschl's gyrus also belonged to the non-differentiated network.

On the other hand, there were five clusters associated with **silent reading** rather than reading aloud (Figure A.2, purple blobs): the left supplementary motor area, inferior frontal gyrus (pars orbitalis), superior frontal gyrus, middle temporal gyrus, and middle occipital gyrus. Among these, the cluster in the left middle temporal gyrus was also significantly associated with the word-related network.

Furthermore, three clusters were significantly associated with the task of **reading** rather than with lexical decision (Figure A.3, green blobs): a cluster in the left cerebellar hemisphere, a cluster in the left middle occipital gyrus, and the one in the right Heschl's gyrus.

Finally, seven clusters (Figure A.3, yellow blobs) were significantly associated with the task of **lexical decision** rather than with reading per se: in the left hemisphere, in the middle frontal gyrus, in the angular gyrus, in the inferior parietal lobule, in the middle temporal gyrus (a different cluster from

[1] Tasks requiring implicit reading (as for instance passive viewing, or case detection) were included in the silent reading category.

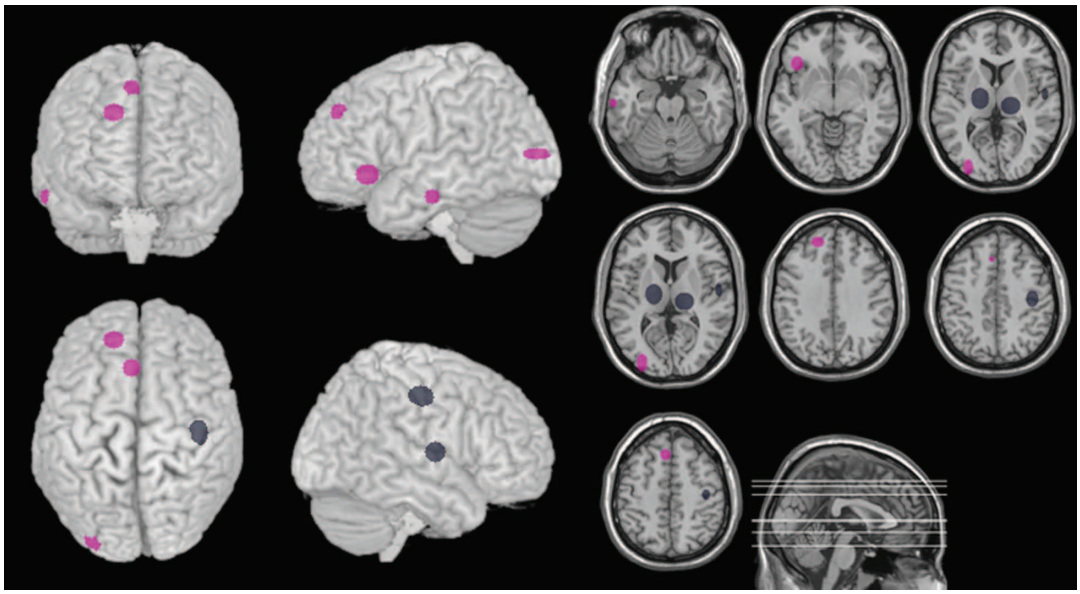


Figure A.2: Clusters significantly associated with reading aloud as opposed to reading silently are shown in dark blue; conversely, clusters significantly associated with silent reading are shown in purple.

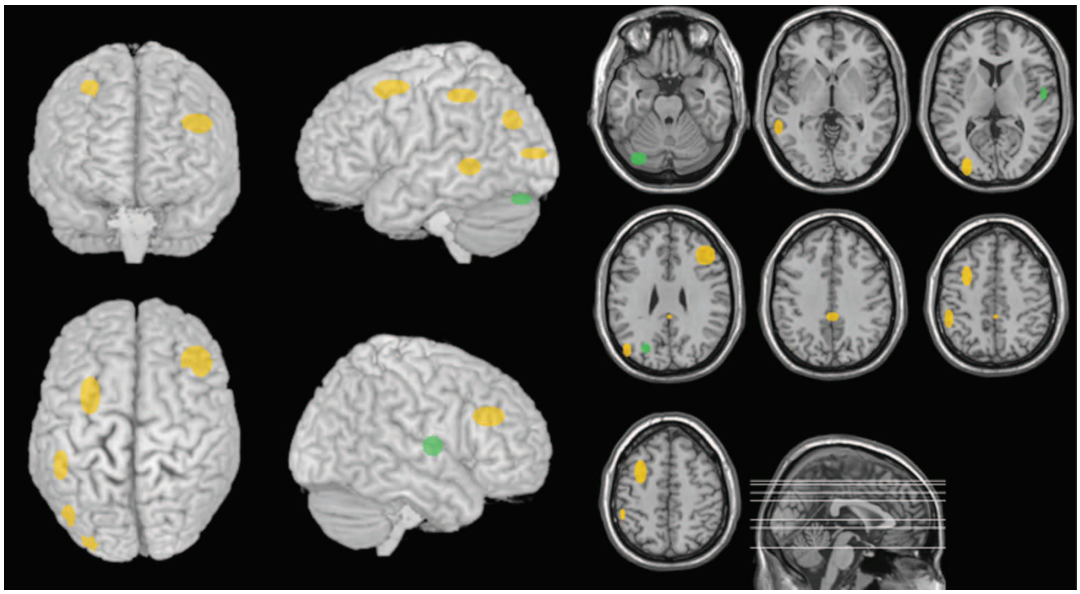


Figure A.3: Yellow blobs represent clusters significantly associated with the task of lexical decision rather than with reading per se; green blobs identify the clusters that were found to be significantly more associated with reading rather than with lexical decision.

the one associated with silent reading), and in the middle occipital gyrus; in the right hemisphere, in the right inferior frontal gyrus (pars triangularis) and in the mid-cingulum. Among these, the clusters in the left angular gyrus and the middle temporal gyrus were also significantly associated with the word-related network. In particular, for the left middle temporal gyrus the association with the word-related network was driven by the activation peaks for the lexical decision task.

Taken together, the findings of the reading aloud vs. reading silently comparison seem to suggest a greater neural work for reading aloud in areas involved in the analysis of re-entrant speech (the right Heschl's gyrus), while reading silently may prove as a more "cognitively demanding" task as indicated by the larger frontal involvement.

The same considerations may apply to the lexical decision task as opposed to reading: higher-order frontal and temporo-parietal cortices are associated with lexical decision while regions involved in speech planning or speech monitoring are more systematically associated with reading *per se*.

A discussion on previous meta-analytic studies on reading

Our meta-analysis has taken inspiration from the work by Jobard et al. (2003), who reconsidered published neuroimaging studies up to 2002 investigating reading processes. Similarly to what we did here, they employed a hierarchical clustering algorithm in order to group close activation peaks, and then they proceeded to analyze the resulting clusters. However, the analytic step was performed in a way that is different under many respects from the approach we have taken here.

The first divergence point lies in the procedure of classification of the statistical comparisons of the experimental condition (usually performed via linear contrast, henceforth *contrasts*), which, in turns, is a direct consequence of different theoretical assumptions. Jobard and colleagues (2003) explicitly shaped their study within a dual-route account of reading processes. Therefore, they classified contrasts according to this model, resulting in four contrast categories: direct (lexicosemantic) route, indirect (GPC) route, non-conclusive, word vs. pictures. In our meta-analysis, we made no attempt to fit contrasts into a specific framework, and just organized them along a lexicality axis, while also considering the hypothesis that a lexicality-independent difficulty effect might arise. Incidentally, this also means that a contrast (e.g. pseudoword $>$ word) is not univocally classified, but can contribute to more categories (e.g. the pseudoword category, and the difficulty one). Our atheoretical classification of the statistical comparisons did not prevent us from discussing the results in the light of competing cognitive models of reading; at the same time, it did not bias the meta-analytic process toward an interpretation that is specific of both the chosen theoretical account, and the investigators' personal understanding of it. For instance, let us consider the classification of the contrast *pseudoword* $>$ *word* in the Indirect Route class, operated by Jobard et al. (2003). The assumption here is that a pseudoword would stress the GPC route more than a word; however, it can be assumed that both words and pseudowords are processed by this route; the fact that words can also benefit from a stored representation in the orthographic lexicon does not prevent the GPC route to also process them. If this is the case, such a contrast would not be very effective in pointing out the GPC route. Similar considerations can be made for the *irregular* $>$ *regular* word contrast, that Jobard and colleagues (2003) classified as a Direct Route one. However, since both regular and irregular words are equally represented in an orthographic lexicon, they both should engage the lexical route to a similar extent.

We were also more strict in our inclusion criteria for studies, statistical comparisons, and activation coordinates, which explains why, even though we considered a larger time span than in (Jobard et al., 2003) study, our final database still contained 35 studies. In particular, we set a strict threshold on

P-values: this led us to discard many activation coordinates, but we considered it to be a necessary operation in order to have a homogeneous and robust (in a statistical sense) dataset.

Of the 55 clusters returned by the clustering procedure, only 11 (two of which were then considered as one in the discussion) were reported in (Jobard et al., 2003). The choice of these eleven clusters was made because they are *"regions in the left hemisphere that have been described previously in the literature as having a role in word reading"*. In practice this may be a sensible decision, yet it involves a certain degree of circularity that may invalidate the whole enterprise of performing a meta-analysis; in addition, such a priori approach may lead to disregard interesting, possibly unexpected, outcomes that might emerge in other brain regions. In our work, we decided not to exclude any of the clusters we found from our analyses, although only for those showing a reliable (lexicality or difficulty) effect we provided an in-depth discussion.

Another major difference between (Jobard et al., 2003) work and our own lies in the way the functional role for each cluster was evaluated. Jobard and colleagues (2003) decided on the role of a cluster by basically counting the number of Direct Route contrasts versus the number of Indirect Route contrasts: when a disproportion was found, the functional assignment was made; otherwise, the role of that cluster would be decided on the basis of previous findings in literature. However, the magnitude (and, therefore, the reliability) of such disproportions was not assessed statistically, leaving space for uncertainty regarding the robustness of the authors' conclusions. Indeed, there were also assignments that were made *despite* the lack of a (even qualitative) disproportion: *"Even though the opercular cluster contained as many "indirect route" as "direct route" contrasts (three in both cases), we chose to insert it as a part of the network involved in the graphophonological conversion, as we think such a claim is supported by additional experimental evidence"*. Other clusters, collecting a similar number of contrasts from both classes, or mainly non-conclusive ones, were deemed to have a role in semantics.

A comparison among the conclusions drawn by Jobard et al. (2003) meta-analysis and our own is now in order, although it is necessarily incomplete due to the a priori selection of clusters they performed (e.g. no right cluster is discussed). There is some agreement in the semantic areas they found (posterior part of the middle and inferior temporal gyrus), although they also included the triangular part of the left inferior frontal gyrus, that we classified as non-differentiated. However, in Jobard et al. (2003) the rationale that determined the assignment of a cluster to semantic processing was similar to the one we adopted for the non-differentiated class: no predominance of any other specific class. Thus, it is hard to tell if the areas they report as semantic ones are indeed semantic, or are seen in reading experiments as a consequence of a given task format. They also identified the superior and middle temporal gyrus, supramarginal gyrus, and opercular part of the inferior frontal gyrus as part of the GPC route. A comparison here is even harder, because we did not directly interpreted our results in terms of components of a dual-route model of reading. In particular, we disagree with the interpretation given to the pars opercularis of the inferior frontal gyrus, that was somewhat made arbitrarily on the basis of previous reports in literature. Assuming a dual-route framework, the confused picture regarding this cluster may suggest that a common process to the two routes is carried out in this area. Combining this observation with our results, there is some support that the left operculum can be considered as a convergence point for the two routes, where the (possibly) competing phonological representations interact and contribute to the final output. Finally, the authors state that no area hosting an orthographic lexicon could be found, and that the direct route must be implemented by the coactivation of regions devoted to prelexical processing (in the occipito-temporal junction) and that of semantic areas. Interestingly, they claim that their results support a dual-route theory of reading, although they did not support one of the main tenets of such theories (the existence of an orthographic

lexicon storing whole-word forms). Still, it is true that the model they assume in their study is not a classical one: by direct route, they refer to a direct access to meaning from orthography, with the indirect route involving an intermediate step of graphophonological conversion.

Finally, let us briefly point out that, although the clustering methodology adopted here is very similar to the one in (Jobard et al., 2003), some differences still exist. First of all, we employed a modified version of the Ward's clustering algorithm, which we developed to the precise aim to deal with the problem of non-uniqueness of the solution (see Materials and methods of the article). To our knowledge, this problem has not been taken into account in (Jobard et al., 2003) meta-analysis; this means that the clustering solution they discussed – and possibly the conclusions they drew based on that – might have changed, if only the input data order was changed. Moreover, we performed an additional processing step on our clustering solution in order to guarantee some degree of anatomical consistency: this prevented strikingly incompatible anatomical regions to be included in the same cluster and, therefore, to be discussed as part of an (implausible) neural population. The insensitivity of spatial clustering techniques to anatomical constraints is a limit of clustering-based meta-analyses that calls for careful evaluation of cluster plausibility by the experimenter; in a future perspective, an extension of the method, that automatically integrates anatomical information into the clustering process, would constitute a relevant improvement.

Lastly, it is worth mentioning that a different meta-analytic technique has also been introduced in the neuroimaging literature: the Activation Likelihood Estimate (ALE)-based method developed by Turkeltaub and collaborators (2002). This approach consists in modelling each activation peak as a 3D Gaussian distribution; altogether, these distributions are used to build a statistical map (an ALE map), where each voxel is assigned the probability of containing at least one point in the activation dataset. Statistical testing is then performed in order to assess the significance of single voxels in the map; significant voxels thus correspond to the most consistent activations across the considered neuroimaging studies. In (Turkeltaub, 2002), the method was applied to data collected from 11 PET studies on aloud single word reading, and found as consistent areas of activation the bilateral motor cortex, pre-SMA, superior temporal sulci, bilateral cerebellum, and left fusiform gyrus. As there was no attempt to differentiate among specific components of the reading network, a comparison with our results is less interesting. As for the methodology, the ALE approach provides an elegant way to determine the level of concordance across studies for considered brain regions, and therefore it is more efficient in washing out less robust results. On the other hand, as the method only provides the set of consistent areas of activation in a given dataset, no evaluation about the differential contributions (in terms of types of stimuli used, experimental tasks, and so on) to the activation of a given brain area is immediately available. An alternative *modus operandi* would be that of creating several datasets, each for a different subprocess of interest, and separately run the meta-analysis on each of them. However, such a priori classification would prevent the potential detection of other effects, which were neglected at first, but could show in a subsequent analysis. For this reason, we chose to adopt the clustering-based meta-analytic approach instead, although we recognize the potentiality and elegance of the ALE method.

The ALE approach, and a similar method named Aggregated Gaussian-Estimated Sources (AGES – Chein et al., 2002), were used in two meta-analyses on reading, comparing alphabetic systems to logographic ones. In (Bolger et al., 2005), consistent findings on single word reading were investigated for western orthographies and eastern systems (Chinese and Japanese), both separately and in an aggregate way. This allowed highlighting commonalities and specificities for these writing systems. Consistent foci of activation for alphabetic systems (as our meta-analysis included only alphabetic

stimuli, we will not discuss results on eastern orthographic systems) were found in the left ventral occipito-temporal region, left superior posterior temporal area, left inferior frontal gyrus, and left insula/premotor cortex. Tan and colleagues (2005) considered both alphabetic systems and Chinese, focusing on phonological tasks. Areas consistently involved in the phonological processing of alphabetic words were found in the left inferior frontal gyrus (both lateral and medial), left supramarginal gyrus, left mid-superior temporal area, left fusiform gyrus, right superior temporal gyrus and mid-inferior occipital gyrus, and in the cerebellum. Similar considerations as above, about the informativeness of comparing such "consistency results" to our findings, hold for these studies as well.

A.4 Notations used in Chapter 4

We have tried to keep a uniform notation style in Chapter 4, although the variety of notations and terminology that is present in the artificial neural networks (ANNs) literature can sometimes be difficult to reconcile with. Networks with no hidden layers are either referred to as single-layer or two-layer networks; in the first case, input nodes are not considered to be actual units of the network, as they are usually clamped to the input value rather than compute an activation function.

However, we believe the two-layer network expression to give a more intuitive feeling of a network architecture, and thus we consistently use that to refer to networks with no hidden units. Input units are indexed with $j = [1, n]$, and output units with $i = [1, m]$. When we are more broadly referring to generic units in a multi-layer network, we typically prefer indexes p, q, k.

The quantities that characterize ANN dynamics are generally:

- the net input to each unit p: net_p;
- the weight of the connection from unit q to unit p: w_{pq};
- the output, or activation level, of each unit p: $y_p = f(net_p)$, where f denotes the activation function of the unit;
- the threshold of each unit p: ϑ_p – the threshold is usually not made explicit in the computation of the net input, but rather is considered as the weight of the connection from fictitious input terminal $j = 0$ to unit p ($\vartheta_p = w_{p0}$);
- the target (desired output) of each unit p: d_p.

We also use the notation x_j to denote the j-th input to the network, to which the corresponding input unit is clamped. This may yield two different notations when giving the computation of the net input to a unit:

$$net_p = \sum_q w_{pq} x_q \quad \text{or} \quad net_p = \sum_q w_{pq} y_q$$

depending on whether unit p gets input directly from the input units or from intermediate units; in the most general case, the second notation is used. Note that when no ambiguities exist, indexes can be dropped (for instance, if we have one output unit only that receives input from n input nodes, connection weights can be denoted by w_j rather than w_{1j}).

Bibliography

Ackerman, M., Ben-David, S. (2009). Measures of Clustering Quality: A Working Set of Axioms for Clustering. In: Advances in Neural Information Processing Systems 21, pp. 121–128.

Ackerman, M., Ben-David, S., Loker, D. (2010). Characterization of Linkage-based Clustering. In: Proceedings of the Conference on Learning Theory, pp. 270–281.

Ackley, D. H., Hinton, G. E., Sejnowski, T. J. (1985). A Learning Algorithm for Boltzmann Machines. Cognitive SCience 9, pp. 147–169.

Adolphs, R. (1999). The human amygdala and emotion. The Neuroscientist 5 (2), pp. 125–137.

Adolphs, R., Cahill, L., Schul, R., Babinsky, R. (1997). Impaired declarative memory for emotional material following bilateral amygdala damage in humans. Learning & Memory 4 (3), pp. 291–300.

Adolphs, R., Tranel, D., Damasio, H., Damasio, A. (1994). Impaired recognition of emotion in facial expressions following bilateral damage to the human amygdala. Nature 372 (6507), pp. 669–672.

Agresti, A. (1992). A survey of exact inference for contigency tables. Statistical Science 7 (1), pp. 131–177.

Amir, N., Ron, S. (1998). Towards an automatic classification of emotions in speech. In: Fifth International Conference on Spoken Language Processing, pp. 555–558.

Anderson, K., McOwan, P. W. (2006). A real-time automated system for the recognition of human facial expressions. IEEE Transactions on Systems, Man and Cybernetics, Part B 36 (1), pp. 96–105.

André, E., Klesen, M., Gebhard, P., Allen, S., Rist, T. (1999). Integrating models of personality and emotions into lifelike characters. In: Proceedings of the Workshop on Affect in Interactions – Towards a new Generation of Interfaces, pp. 136–149.

Andrews, S. (1997). The effect of orthographic similarity on lexical retrieval: Resolving neighborhood conflicts. Psychonomic Bulletin & Review 4 (4), pp. 439–461.

Ans, B., Carbonnel, S., Valdois, S. (1998). A connectionist multiple-trace memory model for polysyllabic word reading. Psychological Review 105 (4), pp. 678–723.

Argyle, M. (1975). Bodily Communication. International Universities Press, Inc.

Arkin, R. C., Fujita, M., Takagi, T., Hasewaga, R. (2003). An ethological and emotional basis for human-robot interaction. Robotics and Autonomous Systems 42 (3-4), pp. 191–201.

Armony, J. L., Servan-Schreiber, D., Romanski, L. M., Cohen, J. D., LeDoux, J. E. (1997). Stimulus generalization of fear responses: effects of auditory cortex lesions in a computational model and in rats. Cerebral Cortex 7 (2), pp. 157–165.

Ashraf, A. B., Lucey, S., Cohn, J. F., Chen, T., Ambadar, Z., Prkachin, K. M., Solomon, P. E. (2009). The painful face – Pain expression recognition using active appearance models. Image and Vision Computing 27 (12), pp. 1788–1796.

Austermann, A., Esau, N., Kleinjohann, L., Kleinjohann, B. (2005). Prosody Based Emotion Recognition for MEXI. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2430–2436.

Baldi, P., Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. Neural Networks 2, pp. 53–58.

Ball, G., Hall, D. (1967). A clustering technique for summarizing multivariate data. Behavioral Science 12, pp. 153–155.

Baron, J., Strawson, C. (1976). Use of orthographic and word-specific knowledge in reading words aloud. Journal of Experimental Psychology: Human Perception and Performance 2, pp. 386–392.

Bates, J., Loyall, A. B., Reilly, W. S. (1992). An Architecture for Action, Emotion, and Social Behavior. Tech. Rep. CMU-CS-92-144, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Bear, M. F., Connors, B. W., Paradiso, M. A. (2001). Neuroscience: Exploring the Brain, 2nd Edition. Lippincott Williams & Wilkins.

Beauregard, M., Chertkow, H., Bub, D., Murtha, S., Dixon, R., Evans, A. (1997). The neural substrate for concrete, abstract, and emotional word lexica: a Positron Emission Tomography study. Journal of Cognitive Neuroscience 9 (4), pp. 441–461.

Beauvois, M., Derouesné, J. (1979). Phonological alexia: Three dissociations. Journal of Neurology, Neurosurgery and Psychiatry 42, pp. 1115–1124.

Bechara, A., Tranel, D., Damasio, H., Adolphs, R., Rockland, C., Damasio, A. R. (1995). Double dissociation of conditioning and declarative knowledge relative to the amygdala and hippocampus in humans. Science 269 (5227), pp. 1115–1118.

Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 5 (2), pp. 157–166.

Berlingeri, M., Cattinelli, I., Traficante, D., Borghese, N. A., Bottini, G., Paulesu, E. (In Preparation). Task-dependent and task-independent age-related effects: evidence from a new quantitative meta-analytic approach.

Besner, D., Smith, M. C. (1992). Basic processes in reading: Is the orthographic depth hypothesis sinking? In: Frost, R., Katz, L. (Eds.), Orthography, phonology, morphology, and meaning. Elsevier, Amsterdam, pp. 45–66.

Besner, D., Twilley, L., McCann, R. S., Seergobin, K. (1990). On the Association Between Connectionism and Data: Are a Few Words Necessary? Psychological Review 97 (3), pp. 432–446.

Bethel, C. L., Murphy, R. R. (2008). Survey of Non-facial/Non-verbal Affective Expressions for Appearance-Constrained Robots. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews 38 (1), pp. 83–92.

Bezdek, J. (1981). Pattern recognition with fuzzy objective function algorithms. Plenum Press.

Bienenstock, E. L., Cooper, L. N., Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. Journal of Neuroscience 2, pp. 32–48.

Binder, J. R., Desai, R. H., Graves, W. W., Conant, L. L. (2009). Where is the semantic system? A critical review and meta-analysis of 120 functional neuroimaging studies. Cerebral Cortex 19 (12), pp. 2767–96.

Binder, J. R., McKiernan, K. A., Parsons, M. E., Westbury, C. F., Possing, E. T., Kaufman, J. N., Buchanan, L. (2003). Neural correlates of lexical access during visual word recognition. Journal of Cognitive Neuroscience 15 (3), pp. 372–393.

Binder, J. R., Medler, D. A., Desai, R., Conant, L. L., Liebenthal, E. (2005). Some neurophysiological constraints on models of word naming. NeuroImage 27 (3), pp. 677–693.

Bird, H., Lambon Ralph, M. A., Seidenberg, M. S., McClelland, J. L., Patterson, K. (2003). Deficits in phonology and past-tense morphology: What's the connection? Journal of Memory and Language 48 (3), pp. 502–526.

Bisiacchi, P. S., Cipolotti, L., Denes, G. (1989). Impairment in processing meaningless verbal material in several modalities: the relationship between short term memory and phonological skills. The Quarterly Journal of Experimental Psychology 41 A (2), pp. 293–319.

Boden, M. A. (2008). An Evaluation of Computational Modeling in Cognitive Science. In: Sun, R. (Ed.), Cambridge handbook of computational psychology. Cambridge University Press, pp. 667–683.

Bolger, D. J., Perfetti, C. A., Schneider, W. (2005). Cross-cultural effect on the brain revisited: universal structures plus writing system variation. Human Brain Mapping 25 (1), pp. 92–104.

Bookheimer, S. Y., Zeffiro, T. A., Blaxton, T., Gaillard, W., Theodore, W. (1995). Regional cerebral blood flow during object naming and word reading. Human Brain Mapping 3, pp. 93–106.

Bosagh Zadeh, R., Ben-David, S. (2009). A uniqueness theorem for clustering. In: Proceedings of the Twenty-Fifth Annual Conference on Uncertainty in Artificial Intelligence (UAI-09), pp. 639–646.

Bottini, G., Paulesu, E. (2003). Functional neuroanatomy of spatial perception, spatial processes, and attention. In: Halligan, P. W., Kischka, U., Marshall, J. C. (Eds.), Handbook of clinical neuropsychology. Oxford University Press, New York, Ch. 38, pp. 697–723.

Botvinick, M. M., Plaut, D. C. (2006). Short-term memory for serial order: a recurrent neural network model. Psychological Review 113 (2), pp. 201–233.

Bower, G. H. (1981). Mood and memory. American Psychologist 36 (2), pp. 129–148.

Bowers, J. S. (2009). On the Biological Plausibility of Grandmother Cells: Implications for Neural Network Theories in Psychology and Neuroscience. Psychological Review 116 (1), pp. 220–251.

Bowers, J. S. (2010). More on grandmother cells and the biological implausibility of PDP models of cognition: a reply to Plaut and McClelland (2010) and Quian Quiroga and Kreiman (2010). Psychological Review 117 (1), pp. 300–308.

Brambati, S. M., Ogar, J., Neuhaus, J., Miller, B. L., Gorno-Tempini, M. L. (2009). Reading disorders in primary progressive aphasia: a behavioral and neuroimaging study. Neuropsychologia 47 (8-9), pp. 1893–1900.

Breazeal, C. (2003). Emotion and sociable humanoid robots. International Journal of Human-Computer Studies 59 (1-2), pp. 119–155.

Breazeal, C. L. (2002). Designing Sociable Robots. MIT Press, Cambridge, MA.

Broca, P. (1861). Loss of speech, chronic softening and partial destruction of the anterior left lobe of the brain. Bulletin de la Société Anthropologique de Paris 2, pp. 235–238.

Broca, P. (1878). Anatomie comparee des circonvolutions cerebrales: Le grand lobe limbique et la scissure limbique dans la serie des mammifères. Revue d'Anthropologie 1, pp. 385–498.

Brodmann, K. (1909). Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues. Johann Ambrosius Barth Verlag, Leipzig.

Bub, D., Cancelliere, A., Kertesz, A. (1985). Whole-word and analytic translation of spelling to sound in a nonsemantic reader. In: Patterson, K. E., Marshall, J. C., Coltheart, M. (Eds.), Surface dyslexia: neuropsychological and cognitive studies of phonological reading. Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 15–34.

Buchel, C., Price, C., Friston, K. (1998). A multimodal language region in the ventral visual pathway. Nature 394, pp. 274–277.

Bullinaria, J. A. (1996). Connectionist Models of Reading: Incorporating Semantics. In: Proceedings of the First European Workshop on Cognitive Modelling, pp. 224–229.

Bullinaria, J. A. (1997). Modeling reading, spelling, and past tense learning with artificial neural networks. Brain and Language 59 (2), pp. 236–266.

Cañamero, D. (1997). Modeling motivations and emotions as a basis for intelligent behavior. In: Proceedings of the first international conference on Autonomous agents - AGENTS '97, pp. 148–155.

Cahn, J. E. (1990). The generation of affect in synthesized speech. Journal of the American Voice I/O Society 8, pp. 1–19.

Calvo, R. A., D'Mello, S. (2010). Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. IEEE Transactions on Affective Computing 1 (1), pp. 18–37.

Campadelli, P., Lanzarotti, R. (2002). Localization of facial features and fiducial points. In: Proceedings of the IASTED International Conference Visualisation, Imaging and Image Processing (VIIP2002), pp. 419–495.

Canamero, L., Fredslund, J. (2001). I show you how I like you – Can you read it in my face? IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans 31 (5), pp. 454–459.

Canli, T., Zhao, Z., Brewer, J., Gabrieli, J. D., Cahill, L. (2000). Event-related activation in the human amygdala associates with later memory for individual emotional experience. The Journal of Neuroscience 20 (RC99), pp. 1–5.

Cannon, W. B. (1927). The james-lange theory of emotions: a critical examination and an alternative theory. American Journal of Psychology 100 (3-4), pp. 567–86.

Cappa, S. F., Vignolo, L. A. (1999). The neurological foundations of language. In: Denes, G., Pizzamiglio, L. (Eds.), Handbook of clinical and experimental neuropsychology. Psychology Press, Ch. 8, pp. 155–179.

Carpenter, G., Grossberg, S. (1988). The art of adaptive pattern recognition by a self-organizing neural network. Computer, pp. 77–88.

Carpenter, G. A., Grossberg, S. (1987a). Art2: Self-organization of stable category recognition codes for analog input patterns. Applied Optics 26, pp. 4919–4930.

Carpenter, G. A., Grossberg, S. (1987b). A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing 37, pp. 54–115.

Carreiras, M., Mechelli, A., Price, C. J. (2006). Effect of word and syllable frequency on activation during lexical decision and reading aloud. Human Brain Mapping 27 (12), pp. 963–972.

Carreiras, M., Riba, J., Vergara, M., Heldmann, M., Münte, T. F. (2009). Syllable congruency and word frequency effects on brain activation. Human Brain Mapping 30 (9), pp. 3079–88.

Carver, C. S. (2006). Approach, avoidance, and the self-regulation of affect and action. Motivation and Emotion 30 (2), pp. 105–110.

Carver, C. S., Scherier, M. F. (1990). Origins and functions of positive and negative affect: A control-process view. Psychological Review 97 (1), pp. 19–35.

Carver, C. S., Scherier, M. F. (1998). On the self-regulation of behavior. Cambridge University Press.

Cattinelli, I. (2006). Interazione Emotiva con Robot AIBO, Unpublished Master's Thesis (in Italian), Università degli Studi di Milano.

Cattinelli, I., Borghese, N. A., Gallucci, M., Paulesu, E. (Under Revision–a). Reading the reading brain: a new meta-analysis of functional imaging data on reading.

Cattinelli, I., Goldwurm, M., Borghese, N. A. (2008). Interacting with an artificial partner: modeling the role of emotional aspects. Biological Cybernetics 99 (6), pp. 473–489.

Cattinelli, I., Valentini, G., Paulesu, E., Borghese, N. A. (Under Revision–b). A novel approach to the problem of non-uniqueness of the solution in hierarchical clustering.

Cavalli-Sforza, L., Edwards, A. (1967). Phylogenetic analysis models and estimation procedures. Amer. J. Human Genetics 19, pp. 233–257.

Chee, M. W., O'Craven, K. M., Bergida, R., Rosen, B. R., Savoy, R. L. (1999). Auditory and visual word processing studied with fMRI. Human Brain Mapping 7 (1), pp. 15–28.

Chein, J. M., Fissell, K., Jacobs, S., Fiez, J. A. (2002). Functional heterogeneity within Broca's area during verbal working memory. Physiology & behavior 77 (4-5), pp. 635–9.

Chernova, S., Arkin, R. C. (2007). From deliberative to routine behaviors: a cognitively inspired action-selection mechanism for routine behavior capture. Adaptive Behavior 15, pp. 199–216.

Chittaro, L., Serra, M. (2004). Behavioral programming of autonomous characters based on probabilistic automata and personality. Computer Animation and Virtual Worlds 15 (3-4), pp. 319–326.

Christiansen, M. H., Chater, N. (Eds.) (2001). Connectionist Psycholynguistics. Ablex Publishing.

Churchland, P., Sejnowski, T. J. (1994). The Computational Brain. MIT Press, Cambridge, MA, USA.

Ciota, Z. (2005). Emotion recognition on the basis of human speech. In: The 18th International Conference on Applied Electromagnetics and Communications, pp. 1–4.

Cohen, L., Dehaene, S., Naccache, L., Lehericy, S., Dehaene-Lambertz, G., Henaff, M. A., Michel, F. (2000). The visual word form area: spatial and temporal characterization of an initial stage of reading in normal subjects and posterior split-brain patients. Brain 123 (Pt 2), pp. 291–307.

Cohen, L., Jobert, A., Le Bihan, D., Dehaene, S. (2004). Distinct unimodal and multimodal regions for word processing in the left temporal cortex. NeuroImage 23 (4), pp. 1256–1270.

Cohen, L., Lehericy, S., Chochon, F., Lemer, C., Rivaud, S., Dehaene, S. (2002). Language-specific tuning of visual cortex? Functional properties of the Visual Word Form Area. Brain 125 (Pt 5), pp. 1054–1069.

Coltheart, M. (1978). Lexical access in simple reading tasks. In: Underwood, G. (Ed.), Strategies of information processing. Academic Press, London, pp. 151–216.

Coltheart, M. (1982). The psycholinguistic analysis of acquired dyslexias: some illustrations. Philosophical Transactions of the Royal Society of London 298 (1089), pp. 151–164.

Coltheart, M. (1985). Cognitive neuropsychology and the study of reading. In: Posner, M. I., Marin, O. S. M. (Eds.), Attention and Performance XI. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, pp. 3–37.

Coltheart, M. (1996). Phonological Dyslexia: Past and Future Issues. Cognitive Neuropsychology 13 (6), pp. 749–762.

Coltheart, M., Curtis, B., Atkins, P., Haller, M. (1993). Models of reading aloud: Dual-route and parallel-distributed-processing approaches. Psychological Review 100 (4), pp. 589–608.

Coltheart, M., Masterson, J., Byng, S., Prior, M., Riddoch, J. (1983). Surface dyslexia. The Quarterly Journal Of Experimental Psychology Section A 35 (3), pp. 469–495.

Coltheart, M., Patterson, K., Marshall, J. C. (1980). Deep dyslexia. Routledge & Kegan Paul, London.

Coltheart, M., Rastle, K. (1994). Serial processing in reading aloud: Evidence for dual-route models of reading. Journal of Experimental Psychology: Human Perception and Performance 20, pp. 1197–1211.

Coltheart, M., Rastle, K., Perry, C., Langdon, R., Ziegler, J. (2001). DRC: a dual route cascaded model of visual word recognition and reading aloud. Psychological Review 108 (1), pp. 204–256.

Corbetta, M. (1998). Frontoparietal cortical networks for directing attention and the eye to visual locations: identical, independent, or overlapping neural systems? Proceedings of the National Academy of Sciences of the United States of America 95 (3), pp. 831–8.

Cormack, R. M. (1971). A Review of Classification. Journal of the Royal Statistical Society. Series A (General) 134 (3), pp. 321–367.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2001). Introduction to Algorithms, 2nd Edition. The MIT Press.

Cortese, M. J. (1998). Revisiting serial position effects in reading. Journal of Memory and Language 39, pp. 652–665.

Costa, P T, J., McCrae, R. R. (1992). Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI) Professional Manual. Psychological Assessment Resources.

Cottrell, G. W., Metcalfe, J. (1991). EMPATH: Face, Emotion, and Gender Recognition Using Holons. In: Advances in Neural Information Processing Systems, pp. 564–571.

Cottrell, G. W., Munro, P., Zipser, D. (1987). Learning internal representations from gray-scale images: An example of extensional programming. In: The Ninth Annual Conference of the Cognitive Science Society, pp. 462–473.

Cottrell, G. W., Munro, P., Zipser, D. (1989). Image compression by back propagation: A demonstration of extensional programming. In: Sharkey, N. (Ed.), Models of Cognition: A Review of Cognitive Science. Ablex, pp. 208–240.

Crepaldi, D., Berlingeri, M., Cattinelli, I., Borghese, N. A., Luzzatti, C., Paulesu, E. (In Preparation). Clustering the lexicon in the brain: a meta-analysis of the neurofunctional evidence on noun and verb processing.

Cybenko, G. (1988). Continuous valued neural networks with two hidden layers are sufficient. Tech. rep., CS Dept., Tufts University, Medford, MA.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems 2, pp. 303–314.

Damasio, A. R. (1994). Descartes' Error: Emotion, Reason, and the Human Brain. G.P. Putnam's Sons.

Damasio, H., Damasio, A. R. (1980). The anatomical basis of conduction aphasia. Brain 103, pp. 337–350.

Damasio, H., Grabowski, T., Frank, R., Galaburda, A. M., Damasio, A. R. (1994). The return of phineas gage: clues about the brain from the skull of a famous patient. Science 264 (5126), pp. 1102–1105.

Darwin, C. (1872). The expression of the emotions in man and animals. John Murray.

Davidson, R. J., Irwin, W. (1999). The functional neuroanatomy of emotion and affective style. Trends in Cognitive Sciences 3 (1), pp. 11–21.

Dayan, P. (1994). Computational modelling. Current opinion in neurobiology 4 (2), pp. 212–217.

Dayan, P. (2003). Levels of analysis in neural modeling. In: Encyclopedia of cognitive science. Nature Publishing Group/Macmillan, London.

Dayan, P., Abbott, L. F. (2001). Theoretical neuroscience. The MIT Press.

De Silva, P. R., Bianchi-Berthouze, N. (2004). Measuring posture features saliency in expressing affective states. In: Proceedings of the 2004 IEEE International Conference on Intelligent Robots and Systems (IROS'04), pp. 2003–2008.

De Silva, P. R., Osano, M., Marasinghe, A., Madurapperuma, A. P. (2006). Towards recognizing emotion with affective dimensions through body gestures. In: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06), pp. 269–274.

Dejerine, J. (1892). Contribution of l'étude anatomo-pathologique et clinique des differentes varietiés de cecité verbale. Mémoires de la Société de Biologie 4, pp. 61–90.

Dellaert, F., Polzin, T., Waibel, A. (1996). Recognizing emotion in speech. In: Fourth International Conference on Spoken Language Processing, pp. 1970–1973.

Démonet, J.-F., Price, C., Wise, R., Frackowiak, R. (1994). Differential activation of right and left posterior sylvian regions by semantic and phonological tasks: a positron-emission tomography study in normal human subjects. Neuroscience Letters 182 (1), pp. 25–28.

Denes, G., Cipolotti, L., Zorzi, M. (1999). Acquired dyslexias and dysgraphias. In: Denes, G., Pizzamiglio, L. (Eds.), Handbook of clinical and experimental neuropsychology. Psychology Press, pp. 289–317.

Devillers, L., Vidrascu, L. (2006). Real-life emotions detection with lexical and paralinguistic cues on Human-Human call center dialogs. In: Ninth International Conference on Spoken Language Processing, pp. 801–804.

D'haeseleer, P. (2005). How does gene expression clustering work? Nature Biotechnology 23, pp. 1499–1501.

Donato, G., Bartlett, M. S., Hager, J. C., Ekman, P., Sejnowski, T. J. (1999). Classifying facial actions. IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (10), pp. 974–989.

Dopazo, J. (2009). Formulating and testing hypotheses in functional genomics. Artificial Intelligence in Medicine 45 (2-3), pp. 97–107.

Doshi, P., Gmytrasiewicz, P. (2004). Towards affect-based approximations to rational planning: A decision-theoretic perspective to emotions. In: Working Notes of the Spring Symposium on Architectures for Modeling Emotion: Cross-Disciplinary Foundations.

Doya, K. (1992). Bifurcations in the learning of recurrent neural networks. In: Proceedings of IEEE International Symposium on Circuits and Systems, Vol. 6, pp. 2777–2780.

Dreyfus, H. L. (1965). Alchemy and artificial intelligence. Tech. Rep. P-3244, Rand Corporation.

Duda, R., Hart, P., Stork, D. (2001). Pattern Classification, 2nd Edition. John Wiley & Sons.

Eagly, A. H., Chaiken, S. (1993). The Psychology of Attitudes. Wadsworth Publishing.

Eibl-Eibesfeldt, I. (1973). The expressive behavior of the deaf-and-blind-born. In: Cranach, M. V., Vine, I. (Eds.), Social Communication and Movement: Studies of Interaction and Expression in Man and Chimpanzee. Academic Press, pp. 163–194.

Ekman, P. (1984). Expression and the nature of emotion. In: Scherer, K., Ekman, P. (Eds.), Approaches to Emotion. Erlbaum, pp. 319–344.

Ekman, P. (1992a). Are there basic emotions? Psychological Review 99, pp. 550–553.

Ekman, P. (1992b). An argument for basic emotions. Cognition and Emotion 6 (3-4), pp. 169–200.

Ekman, P., Friesen, W. V. (1978). Manual for the Facial Action Coding System. Consulting Psychologists Press, Inc.

El-nasr, M. S., Yen, J., Ioerger, T. R. (2000). FLAME – Fuzzy Logic Adaptive Model of Emotions. Autonomous Agents and Multi-Agent Systems 3 (3), pp. 219–257.

Elman, J. L. (1990). Finding structure in time. Cognitive Science 14, pp. 179–211.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. Cognitive Science 48, pp. 71–99.

Erwin, E., Obermayer, K., Schulten, K. (1992). Self-organizing maps: ordering, convergence properties and energy functions. Biological Cybernetics 67, pp. 47–55.

Essa, I. A., Pentland, A. P. (1997). Coding, analysis, interpretation, and recognition of facial expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7), pp. 757–763.

Fabri, M., Moore, D., Hobbs, D. (2004). Mediating the expression of emotion in educational collaborative virtual environments: an experimental study. Virtual Reality 7 (2), pp. 66–81.

Fabri, M., Moore, D. J., Hobbs, D. J. (1999). The emotional avatar: Non-verbal communication between inhabitants of collaborative virtual environments. In: Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction (GW '99), pp. 269–273.

Fahlman, S. E., Lebiere, C. (1990). The cascade correlation learning architecture. In: Touretzky, D. (Ed.), Advances in Neural Information Processing Systems, Vol. 2, pp. 524–532.

Falcon, S., Gentleman, R. (2007). Using gostats to test gene lists for go term association. Bioinformatics 23 (2), pp. 257–258.

Fasel, B., Luettin, J. (2003). Automatic facial expression analysis: a survey. Pattern Recognition 36 (1), pp. 259–275.

Ferrand, L., New, B. (2003). Syllabic length effects in visual word recognition and naming. Acta Psychologica 113 (2), pp. 167–183.

Ferrari, S., Ferrigno, G., Piuri, V., Borghese, N. A. (2007). Reducing and Filtering Point Clouds with Enhanced Vector Quantization. IEEE Transactions on Neural Networks 18, pp. 161–177.

Fiebach, C. J., Friederici, A. D., Muller, K., von Cramon, D. Y. (2002). fMRI evidence for dual routes to the mental lexicon in visual word recognition. Journal of Cognitive Neuroscience 14 (1), pp. 11–23.

Fiebach, C. J., Ricker, B., Friederici, A. D., Jacobs, A. M. (2007). Inhibition and facilitation in visual word recognition: prefrontal contribution to the orthographic neighborhood size effect. NeuroImage 36 (3), pp. 901–911.

Fiez, J. A. (1997). Phonology, semantics, and the role of the left inferior prefrontal cortex. Human Brain Mapping 5 (2), pp. 79–83.

Fiez, J. A., Balota, D. A., Raichle, M. E., Petersen, S. E. (1999). Effects of lexicality, frequency, and spelling-to-sound consistency on the functional anatomy of reading. Neuron 24 (1), pp. 205–218.

Fiez, J. A., Petersen, S. E. (1998). Neuroimaging studies of word reading. Proceedings of the National Academy of Sciences of the United States of America 95 (3), pp. 914–921.

Fiez, J. A., Tranel, D., Seager-frerichs, D., Damasio, H. (2006). Specific reading and phonological processing deficits are associated with damage to the left frontal operculum. Cortex 42, pp. 624–643.

Fitch, W. M., Margoliash, E. (1967). Construction of phylogenetic trees. Science 155, pp. 279–284.

Fong, T., Nourbakhsh, I., Dautenhahn, K. (2003). A survey of socially interactive robots. Robotics and Autonomous Systems 42 (3-4), pp. 143–166.

Forgy, E. (1965). Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. Biometrics 21, pp. 768–780.

Forster, K., Chambers, S. (1973). Lexical access and naming time. Journal of Verbal Learning and Verbal Behavior 12 (6), pp. 627–635.

Forster, K. I. (1976). Accessing the mental lexicon. In: Wales, R. J., Walker, E. C. T. (Eds.), New approaches to the language mechanisms. North-Holland, Amsterdam.

Frean, M. (1990). The upstart algorithm: a method for constructing and training feedforward neural networks. Neural Computation 2 (2), pp. 198–209.

Frean, M. (2003). Connectionist architectures: Optmization. In: Nadel, L. (Ed.), Encyclopedia of cognitive science. Nature Publishing Group/Macmillan.

Frederiksen, J. R., Kroll, J. F. (1976). Spelling and sound: Approaches to the internal lexicon. Journal of Experimental Psychology: Human Perception and Performance 2, pp. 361–379.

Friedman, R. B. (1996). Recovery from deep alexia to phonological alexia. Brain and Language 52, pp. 114Ű–128.

Frijda, N. H. (1986). The Emotions. Cambridge University Press.

Friston, K. J., Ashburner, J. T., Kiebel, S. J., Nichols, T. E., Penny, W. D. (Eds.) (2007). Statistical Parametric Mapping: The Analysis of Functional Brain Images. Elsevier, London.

Friston, K. J., Frith, C., Liddle, P. F., Dolan, R., Lammertsma, A. A., Frackowiak, R. S. J. (1990). The relationship between global and local changes in PET scans. Journal of Cerebral Blood Flow and Metabolism 10, pp. 458–466.

Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J. B., Frith, C. D., Frackowiak, R. S. J. (1995). Statistical parametric maps in functional imaging: A general linear approach. Human Brain Mapping 2, pp. 189–210.

Fritzke, B. (1995). A growing neural gas network learns topologies. In: Advances in Neural Information Processing Systems 7, pp. 625–632.

Funnell, E. (1983). Phonological processes in reading: New evidence from acquired dyslexia. British Journal of Psychology 74, pp. 159–180.

Gadanho, S. C., Hallam, J. (2001). Robot Learning Driven by Emotions. Adaptive Behavior 9 (1), pp. 42–64.

Gates, L., Yoon, M. G. (2005). Distinct and shared cortical regions of the human brain activated by pictorial depictions versus verbal descriptions: an fMRI study. NeuroImage 24 (2), pp. 473–486.

Gentleman, R. (2004). Using GO for statistical analysis. In: Antoch, J. (Ed.), Compstat 2004, Proc. in Computational Statistics, Physica-Verlag, Heidelberg, pp. 171–180.

Glosser, G., Friedman, R. B. (1990). The continuum of deep/phonological alexia. Cortex 26, pp. 343–Ű359.

Glushko, R. J. (1979). The organization and activation of orthographic knowledge in reading aloud. Journal of Experimental Psychology: Human Perception and Performance 5, pp. 674–691.

Goldstone, R. L. (1994). The role of similarity in categorization: providing a groundwork. Cognition 52 (2), pp. 125–157.

Gratch, J., Marsella, S. (2004). A domain-independent framework for modeling emotion. Cognitive Systems Research 5 (4), pp. 269–306.

Graves, W. W., Desai, R., Humphries, C., Seidenberg, M. S., Binder, J. R. (2010). Neural Systems for Reading Aloud: A Multiparametric Approach. Cerebral Cortex 20 (8), pp. 1799–815.

Gray, P. O. (2001). Psychology, 4th Edition. Worth Publishers.

Griffiths, T. L., Kemp, C., Tenenbaum, J. B. (2008). Bayesian models of cognition. In: Sun, R. (Ed.), Cambridge handbook of computational psychology. Cambridge University Press, pp. 59–100.

Gross, C. G. (2002). Genealogy of the "Grandmother Cell"İ. The Neuroscientist 8 (5), pp. 512–518.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding. Biological Cybernetics 23, pp. 121–134.

Häggström, . (2002). Finite Markov Chains and Algorithmic Applications. Cambridge University Press.

Hagoort, P., Indefrey, P., Brown, C., Herzog, H., Steinmetz, H., Seitz, R. J. (1999). The neural circuitry involved in the reading of German words and pseudowords: A PET study. Journal of Cognitive Neuroscience 11 (4), pp. 383–398.

Harary, F. (1969). Graph Theory. Addison-Wesley.

Harlow, J. M. (1848). Passage of an iron rod through the head. Boston Medical and Surgical Journal 39, pp. 389–393.

Harlow, J. M. (1868). Recovery from the passage of an iron bar through the head. Bulletin of the Massachusetts Medical Society 2, pp. 327–347.

Harm, M. W., Seidenberg, M. S. (2001). Are There Orthographic Impairments in Phonological Dyslexia? Cognitive Neuropsychology 18 (1), pp. 71–92.

Harm, M. W., Seidenberg, M. S. (2004). Computing the Meanings of Words in Reading: Cooperative Division of Labor Between Visual and Phonological Processes. Psychological Review 111 (3), pp. 662–720.

Haugeland, J. (1985). Artificial intelligence: The very idea. MIT press.

Hauk, O., Davis, M. H., Pulvermüller, F. (2008). Modulation of brain activity by multiple lexical and word form variables in visual word recognition: A parametric fMRI study. NeuroImage 42 (3), pp. 1185–1195.

Haykin, S. (1999). Neural Networks: A Comprehensive Foundation, 2nd Edition. Prentice Hall.

Haynes, J. D., Rees, G. (2006). Decoding mental states from brain activity in humans. Nature Reviews Neuroscience 7, pp. 523–534.

Hebb, D. O. (1949). The organization of behavior: A neuropsychological theory. Wiley.

Herbster, A. N., Mintun, M. A., Nebes, R. D., Becker, J. T. (1997). Regional cerebral blood flow during word and nonword reading. Human Brain Mapping 5 (2), pp. 84–92.

Hertz, J., Krogh, A., Palmer, R. G. (1991). Introduction to the Theory of Neural Computation. Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley.

Hillis, A. E., Newhart, M., Heidler, J., Barker, P., Herskovits, E., Degaonkar, M. (2005). The roles of the "visual word form area" in reading. NeuroImage 24, pp. 548 – 559.

Hinton, G. E. (1986). Learning distributed representations of concepts. In: Proceedings of the Eighth Annual Conference of the Cognitive Science Society, pp. 1–12.

Hinton, G. E. (1989). Connectionist learning procedures. Artificial Intelligence 40, pp. 185–234.

Hinton, G. E., Sejnowski, T. J. (1983). Optimal perceptual inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 448–453.

Hinton, G. E., Sejnowski, T. J. (1986). Learning and relearning in boltzmann machines. In: McClelland, J. L., Rumelhart, D. E., the PDP Research Group (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition. Vol. I. MIT Press, pp. 282–317.

Hinton, G. E., Shallice, T. (1991). Lesioning an Attractor Network: Investigations of Acquired Dyslexia. Psychological Review 98 (1), pp. 74–95.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.

Hopcroft, J. E., Ullman, J. D. (1979). Introduction to automata theory, languages and computation. Addison-Wesley.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the National Academy of Sciences USA, Vol. 79, pp. 2554–2558.

Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks 2, pp. 359–366.

Houghton, G., Zorzi, M. (2003). Normal and impaired spelling in a connectionist dual-route architecture. Cognitive Neuropsychology 20, pp. 115–162.

Howard, D., Patterson, K., Wise, R., Brown, W. D., Friston, K., Weiller, C., Frackowiak, R. (1992). The cortical localization of the lexicons. Positron emission tomography evidence. Brain 115 (6), pp. 1769–1782.

Huber, R., Batliner, A., Buckow, J., Nöth, E., Warnke, V., Niemann, H. (2000). Recognition of emotion in a realistic dialogue scenario. In: Sixth International Conference on Spoken Language Processing, pp. 665–668.

Inamura, T., Inaba, M., Inoue, H. (2004). Pexis: Probabilistic experience representation based adaptive interaction system for personal robots. Systems and Computers in Japan 35 (6), pp. 98–109.

Ingram, R. E. (1984). Toward an information-processing analysis of depression. Cognitive Therapy and Research 8 (5), pp. 443–478.

Invernizzi, P., Gandola, M., Verardi, M., Cattinelli, I., Bottini, G., Borghese, N. A., Paulesu, E. (In Preparation). Anatomo-functional correlates of motor imagery: a quantitative approach to the meta-analysis of neurofunctional evidence in simple motor movements.

Iosifescu, M. (1980). Finite Markov Processes and Their Applications. Wiley.

Jacobs, A. M., Grainger, J. (1994). Models of visual word recognition: Sampling the state of the art. Journal of Experimental Psychology: Human Perception and Performance 20, pp. 1311–1334.

Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. Neural Networks 1, pp. 295–307.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. Tech. Rep. GMD Report 148, German National Research Center for Information Technology.

Jaimes, A., Sebe, N. (2007). Multimodal human-computer interaction: A survey. Computer Vision and Image Understanding 108 (1-2), pp. 116–134.

Jain, A., Dubes, R. (1988). Algorithms for clustering data. Prentice Hall.

Jain, A. K., Murty, M. N., Flynn, P. J. (1999). Data clustering: a review. ACM Computing Surveys 31 (3), pp. 264–323.

James, W. (1884). What is an emotion? Mind 9 (34), pp. 188–205.

Jared, D. (1997). SpellingŰsound consistency affects the naming of high-frequency words. Journal of Memory and Language 36, pp. 505–529.

Jared, D. (2002). Spelling-sound consistency and regularity effects in word naming. Journal of Memory and Language 46, pp. 723–750.

Jared, D., McRae, K., Seidenberg, M. S. (1990). The basis of consistency effects in word naming. Journal of Memory and Language 29, pp. 687Ű–715.

Jessen, F., Erb, M., Klose, U., Lotze, M., Grodd, W., Heun, R. (1999). Activation of human language processing brain regions after the presentation of random letter strings demonstrated with event-related functional magnetic resonance imaging. Neuroscience Letters 270 (1), pp. 13–16.

Jiang, D., Tang, C., Zhang, A. (2004). Cluster analysis for gene expression data: A survey. IEEE Transactions on Knowledge and Data Engineering 16 (11), pp. 1370–1386.

Jobard, G., Crivello, F., Tzourio-Mazoyer, N. (2003). Evaluation of the dual route theory of reading: a metanalysis of 35 neuroimaging studies. NeuroImage 20 (2), pp. 693–712.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In: Proceedings of the Eighth Annual Conference of the Cognitive Science Society, pp. 531–546.

Joubert, S., Beauregard, M., Walter, N., Bourgouin, P., Beaudoin, G., Leroux, J. M., Karama, S., Lecours, A. R. (2004). Neural correlates of lexical and sublexical processes in reading. Brain and Language 89 (1), pp. 9–20.

Kastner, S., Ungerleider, L. (2000). Mechanisms of visual attention in the human cortex. Annual Review of Neuroscience 23, pp. 315–341.

Katz, D. (1960). The functional approach to the study of attitudes. The Public Opinion Quarterly 24 (2), pp. 163–204.

Kaufman, L., Rousseeuw, P. (1990). Finding groups in data: An introduction to cluster analysis. John Wiley & Sons.

Kay, J., Marcel, A. (1981). One process, not two, in reading aloud: Lexical analogies do the work of non-lexical rules. The Quarterly Journal Of Experimental Psychology 33A, pp. 397–413.

Kello, C. T. (2003). The emergence of a double dissociation in the modulation of a single control parameter in a nonlinear dynamical system. Cortex 39 (1), pp. 132–134.

Kello, C. T. (2006). Considering the junction model of lexical processing. In: Andrews, S. (Ed.), From Inkmarks to Ideas: Current Issues in Lexical Processing. Taylor and Francis, New York, NJ.

Kello, C. T., Plaut, D. C. (2000). Strategic control in word reading: evidence from speeded responding in the tempo-naming task. Journal of experimental psychology. Learning, memory, and cognition 26 (3), pp. 719–50.

Kello, C. T., Plaut, D. C. (2003). Strategic control over rate of processing in word reading : A computational investigation. Journal of Memory and Language 48, pp. 207–232.

Kello, C. T., Plaut, D. C., MacWhinney, B. (2000). The task dependence of staged versus cascaded processing: an empirical and computational study of Stroop interference in speech production. Journal of experimental psychology: General 129 (3), pp. 340–60.

Kello, C. T., Sibley, D. E., Plaut, D. C. (2005). Dissociations in Performance on Novel Versus Irregular Items: Single-Route Demonstrations With Input Gain in Localist and Distributed Models. Cognitive Science 29 (4), pp. 627–654.

Khatri, P., Draghici, S. (2005). Ontological analysis of gene expression data: current tools, limitations, and open problems. Bioinformatics 21 (18), pp. 3587–3595.

Kiehl, K. A., Liddle, P. F., Smith, A. M., Mendrek, A., Forster, B. B., Hare, R. D. (1999). Neural pathways involved in the processing of concrete and abstract words. Human Brain Mapping 7 (4), pp. 225–233.

Kirkpatrick, S., Gelatt, C. D. J., Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science 220 (4598), pp. 671–680.

Kleinberg, J. (2003). An Impossibility Theorem for Clustering. In: Advances in Neural Information Processing Systems 15, pp. 463–470.

Kleinsmith, A., Bianchi-Berthouze, N. (2007). Recognizing affective dimensions from body posture. In: Proceedings of the 2nd International Conference on Affective Computing and Intelligent Interaction, pp. 48–58.

Kleinsmith, A., Fushimi, T., Bianchi-Berthouze, N. (2005). An incremental and interactive affective posture recognition system. In: International Workshop on Adapting the Interaction Style to Affective Factors.

Kluver, H., Bucy, P. C. (1937). Psychic blindness and other symptoms following bilateral temporal lobectomy in rhesus monkeys. American Journal of Physiology 119, pp. 352–353.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. Biological Cybernetics 43, pp. 59–69.

Kohonen, T. (1990). The self-organizing map. Proceedings of the IEEE 78 (9), pp. 1464–1480.

Kopecek, I. (2003). Constructing personality model from observed communication. In: Proceedings of the 9th International Conference on User Modeling – Assessing and Adapting to User Attitudes and Affect: Why, When and How?, pp. 28–30.

Krishnapuram, R., Keller, J. M. (1993). A possibilistic approach to clustering. IEEE Transactions on Fuzzy Systems 1 (2), pp. 98–110.

Kronbichler, M., Klackl, J., Richlan, F., Schurz, M., Staffen, W., Ladurner, G., Wimmer, H. (2009). On the functional neuroanatomy of visual word processing: effects of case and letter deviance. Journal of Cognitive Neuroscience 21 (2), pp. 222–229.

Kuchinke, L., Jacobs, A. M., Grubich, C., Vo, M. L., Conrad, M., Herrmann, M. (2005). Incidental effects of emotional valence in single word processing: an fMRI study. NeuroImage 28 (4), pp. 1022–1032.

Kuhnlenz, K., Buss, M. (2004). Towards an emotion core based on a hidden markov model. In: Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN 2004), pp. 119–124.

Kučera, H., Francis, W. N. (1967). Computational Analysis of Present Day American English. Brown University Press, Providence.

Lance, G. N., Williams, W. T. (1966). A generalized sorting strategy for computer classifications. Nature 212, pp. 218–218.

Lance, G. N., Williams, W. T. (1967). A general theory of classificatory sorting strategies. i. hierarchical systems. Computer Journal 9, pp. 373–380.

Landis, T. (2006). Emotional words: what's so different from just words? Cortex 42 (6), pp. 823–830.

Lanitis, a., Taylor, C., Cootes, T. (1997). Automatic interpretation and coding of face images using flexible models. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7), pp. 743–756.

Lanitis, A., Taylor, C. J., Cootes, T. F. (1995). A unified approach to coding and interpreting face images. In: Proceedings of IEEE International Conference on Computer Vision, pp. 368–373.

Lazarus, R. S. (1991). Emotion & Adaptation. Oxford University Press.

Le Cun, Y., Boser, B., Denken, J. S., Henderson, D., Howard, R. E., Hubbard, W., D., J. L. (1989). Back-propagation applied to handwritten zip code recognition. Neural Computation 1, pp. 541–551.

LeCun, Y. (1985). Une procédure d'apprentissage pour réseau a seuil asymmetrique. In: Proceedings of Cognitiva 85, Paris, France, pp. 599–604.

LeCun, Y., Denker, J. S., Solla, S. (1990). Optimal brain damage. In: Touretzky, D. (Ed.), Advances in Neural Information Processing Systems, Vol. 2.

LeDoux, J. E. (1996). The Emotional Brain. Simon & Schuster.

LeDoux, J. E. (2000). Emotion Circuits in the Brain. Annual Review of Neuroscience 23, pp. 155–184.

Lee-Johnson, C. P., Carnegie, D. A. (2010). Mobile Robot Navigation Modulated by Artificial Emotions. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 40 (2), pp. 469–480.

Lester, J. C., Towns, S. G., Callaway, C. B., Voerman, J. L., FitzGerald, P. J. (2000). Deictic and emotive communication in animated pedagogical agents. In: Cassell, J., Sullivan, J., Prevost, S., Churchill, E. F. (Eds.), Embodied conversational agents. MIT Press, Cambridge, MA, USA, pp. 123–154.

Levy, J., Pernet, C., Treserras, S., Boulanouar, K., Aubry, F., Démonet, J.-F., Celsis, P. (2009). Testing for the dual-route cascade reading model in the brain: an fMRI effective connectivity account of an efficient reading style. PloS ONE 4 (8), pp. e6675.

Levy, J., Pernet, C., Treserras, S., Boulanouar, K., Berry, I., Aubry, F., Demonet, J.-F., Celsis, P. (2008). Piecemeal recruitment of left-lateralized brain areas during reading: a spatio-functional account. NeuroImage 43, pp. 581–91.

Litman, D. J., Forbes-Riley, K. (2004). Predicting student emotions in computer-human tutoring dialogues. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04, pp. 351–358.

Logothetis, N. K., Pfeuffer, J. (2004). On the nature of the BOLD fMRI contrast mechanism. Magnetic resonance imaging 22 (10), pp. 1517–1531.

Lukoševičius, M., Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. Computer Science Review 3 (3), pp. 127–149.

Maass, W., Natschläger, T., Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Computation 14 (11), pp. 2531–2560.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium, Vol. 1, pp. 281–297.

Magnenat-Thalmann, N., Primeau, E., Thalmann, D. (1988). Abstract muscle action procedures for human face animation. The Visual Computer 3 (5), pp. 290–297.

Mani, J., Diehl, B., Piao, Z., Schuele, S. S., Lapresto, E., Liu, P., Nair, D. R., Dinner, D. S., Lüders, H. O. (2008). Evidence for a basal temporal visual language center: cortical stimulation producing pure alexia. Neurology 71, pp. 1621–1627.

Marcus, G. F., Pinker, S., Ullman, M., Hollander, M., Rosen, T. J., Xu, F. (1992). Overregularization in language acquisition. Monographs of the Society for Research in Child Development 57 (4).

Marinier, R. P., Laird, J. E., Lewis, R. L. (2009). A computational unification of cognitive behavior and emotion. Cognitive Systems Research 10 (1), pp. 48–69.

Marr, D. (1982). Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. W.H. Freeman & Company.

Marsella, S., Gratch, J. (2009). EMA: A process model of appraisal dynamics. Cognitive Systems Research 10 (1), pp. 70–90.

Marshall, J. C., Newcombe, F. (1973). Patterns of paralexia: A psycholinguistic approach. Journal of Psycholinguistic Research 2 (3), pp. 175–199.

Marslen-Wilson, W. (1987). Functional parallelism in spoken word-recognition. Cognition 25 (1-2), pp. 71–102.

Martinetz, T. M., Berkovich, S. G., Schulten, K. J. (1993). "Neural-gas" network for vector quantization and its application to time-series prediction. IEEE Transactions on Neural Networks 4 (4), pp. 558–569.

Matthews, G., Harvey, T. A. (1996). Connectionist models of emotional distress and attentional bias. Cognition & Emotion 10 (6), pp. 561–600.

Matthews, G., Zeidner, M., Roberts, R. D. (2003). Emotional Intelligence: Science and Myth. The MIT Press.

McCann, R. S., Besner, D. (1987). Reading pseudohomophones: Implications for models of pronunciation assembly and the locus of word-frequency effects in naming. Journal of Experimental Psychology: Human Perception and Performance 13 (1), pp. 14–24.

McCarthy, R., Warrington, E. K. (1986). Phonological reading: Phenomena and paradoxes. Cortex 22, pp. 359–380.

McClelland, J. L. (2009). The Place of Modeling in Cognitive Science. Topics in Cognitive Science 1 (1), pp. 11–38.

McClelland, J. L., Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. an account of basic findings. Psychological Review 88 (5), pp. 375–407.

McClelland, J. L., Rumelhart, D. E., the PDP Research Group (Eds.) (1986). Parallel distributed processing: Explorations in the microstructure of cognition. MIT Press.

McCulloch, W. S., Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, pp. 115–133.

Mechelli, A., Crinion, J. T., Long, S., Friston, K. J., Lambon Ralph, M. A., Patterson, K., McClelland, J. L., Price, C. J. (2005). Dissociating reading processes on the basis of neuronal interactions. Journal of Cognitive Neuroscience 17 (11), pp. 1753–1765.

Mechelli, A., Friston, K. J., Price, C. J. (2000). The effects of presentation rate during word and pseudoword reading: a comparison of PET and fMRI. Journal of Cognitive Neuroscience 12 Suppl 2, pp. 145–156.

Mechelli, A., Gorno-Tempini, M. L., Price, C. J. (2003). Neuroimaging studies of word and pseudoword reading: consistencies, inconsistencies, and limitations. Journal of Cognitive Neuroscience 15 (2), pp. 260–71.

Menard, M. T., Kosslyn, S. M., Thompson, W. L., Alpert, N. M., Rauch, S. L. (1996). Encoding words and pictures: a positron emission tomography study. Neuropsychologia 34 (3), pp. 185–194.

Meschyan, G., Hernandez, A. E. (2006). Impact of language proficiency and orthographic transparency on bilingual word reading: an fMRI investigation. NeuroImage 29 (4), pp. 1135–1140.

Minsky, M. L., Papert, S. A. (1969). Perceptrons. MIT Press.

Mischel, W., Shoda, Y. (1995). A cognitive-affective system theory of personality: reconceptualizing situations, dispositions, dynamics, and invariance in personality structure. Psychological Review 102 (2), pp. 246–268.

Moody, J., Darken, C. (1988). Learning with localized receptive fields. In: Touretzky, D., Hinton, G., Sejnowski, T. (Eds.), Proceedings of the 1988 Connectionist Models Summer School, pp. 133–143.

Moody, J., Darken, C. (1989). Fast learning in networks of locally-tuned processing units. Neural Computation 1, pp. 281–294.

Moore, C. J., Price, C. J. (1999). Three distinct ventral occipitotemporal regions for reading and object naming. NeuroImage 10 (2), pp. 181–192.

Morgan, B. J. T., Ray, A. P. G. (1995). Non-uniqueness and inversions in cluster analysis. Applied Statistics 44 (1), pp. 117–134.

Morrone-Strupinsky, J. V., Lane, R. D. (2003). Neural basis of emotion. In: Nadel, L. (Ed.), Encyclopedia of cognitive science. Nature Publishing Group/Macmillan.

Morton, J. (1969). Interaction of information in word recognition. Psychological Review 76 (2), pp. 165–178.

Morton, J., Patterson, K. (1980). A new attempt at an interpretation, or, an attempt at a new interpretation. In: Coltheart, M., Patterson, K., Marshall, J. C. (Eds.), Deep dyslexia. Routledge & Kegan Paul, London, pp. 91–118.

Munro, P. (2003). Backpropagation. In: Nadel, L. (Ed.), Encyclopedia of cognitive science. Nature Publishing Group/Macmillan.

Mur, M., Bandettini, P., Kriegeskorte, N. (2009). Revealing representational content with pattern-information fMRI–an introductory guide. Social cognitive and affective neuroscience 4 (1), pp. 101–9.

Murtagh, F. (1983). A Survey of Recent Advances in Hierarchical Clustering Algorithms. The Computer Journal 26 (4), pp. 354–359.

Nerb, J., Spada, H. (2001). Evaluation of environmental problems: A coherence model of cognition and emotion. Cognition & Emotion 15 (4), pp. 521–551.

Newell, A., Simon, H. A. (1963). Gps, a program that simulates human thought. In: Computers & Thought. AAAI Press/The MIT Press, pp. 279–293.

Newell, A., Simon, H. A. (1976). Computer Science as Empirical Inquiry: Symbols and Search. Communications of the ACM 19 (3), pp. 113–126.

Niedenthal, P. M. (2003). Emotion. In: Nadel, L. (Ed.), Encyclopedia of cognitive science. Nature Publishing Group/Macmillan.

Nomura, T. (1996). Generation of relations between individuals based on a stochastic automaton and an analogy from social psychology. In: Proc. ALIFE V Poster Presentations, pp. 125–132.

Norman, K. A., Polyn, S. M., Detre, G. J., Haxby, J. V. (2006). Beyond mind-reading: multi-voxel pattern analysis of fMRI data. Trends in Cognitive Sciences 10 (9), pp. 424–430.

Nosarti, C., Mechelli, A., Green, D. W., Price, C. J. (2010). The impact of second language learning on semantic and nonsemantic first language reading. Cerebral Cortex 20 (2), pp. 315–327.

Oatley, K., Johnson-Laird, P. N. (1987). Towards a cognitive theory of emotions. Cognition & Emotion 1 (1), pp. 29–50.

Oatley, K., Johnson-Laird, P. N. (1996). The communicative theory of emotions: Empirical tests, mental models, and implications for social interaction. In: Martin, L. L., Tesser, A. (Eds.), Striving and feeling: Interactions among goals, affect, and self-regulation. Lawrence Erlbaum Associates, pp. 363–393.

Oja, E. (1982). A simplified neuron model as a principal component analyser. Journal of Mathematical Biology 15, pp. 267–73.

Ollman, R. T., Billington, M. J. (1972). The Deadline model for simple reaction times. Cognitive Psychology 3 (2), pp. 311–336.

O'Reilly, R. (1996). Biologically plausible error-driven learning using local activation differences: The Generalized Recirculation algorithm. Neural Computation 8 (5), pp. 895–938.

O'Reilly, R. C., Munakata, Y. (2000). Computational explorations in cognitive neuroscience. MIT Press.

Ortony, A., Clore, G. L., Collins, A. (1988). The Cognitive Structure of Emotion. Cambridge University Press, Cambridge, UK.

Oudeyer, P.-Y. (2003). The production and recognition of emotions in speech: features and algorithms. International Journal of Human-Computer Studies 59 (1-2), pp. 157–183.

Paap, K. R., Noel, R. W. (1991). Dual route models of print to sound: Still a good horse race. Psychological Research 53, pp. 13–24.

Pachella, R. G., Pew, R. W. (1968). Speed-Accuracy Tradeoff in Reaction Time: Effect of Discrete Criterion Times. Journal of Experimental Psychology 76 (1, Pt.1), pp. 19–24.

Padgett, C., Cottrell, G. W. (1997). Representing face images for emotion classification. In: Advances in Neural Information Processing Systems.

Pantic, M., Rothkrantz, L. J. M. (2000). Automatic analysis of facial expressions: the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12), pp. 1424–1445.

Pantic, M., Rothkrantz, L. J. M. (2003). Toward an affect-sensitive multimodal human-computer interaction. Proceedings of the IEEE 91 (9), pp. 1370–1390.

Pantic, M., Sebe, N., Cohn, J. F., Huang, T. (2005). Affective multimodal human-computer interaction. In: 13th Annual ACM International Conference on Multimedia, pp. 669–676.

Papez, J. W. (1937). A proposed mechanism of emotion. Archives of Neurology & Psychiatry 38 (4), pp. 725–743.

Park, K.-H., Lee, H.-E., Kim, Y., Bien, Z. Z. (2008). A steward robot for human-friendly human-machine interaction in a smart house environment. IEEE Transactions on Automation Science and Engineering 5 (1), pp. 21–25.

Parker, D. B. (1982). Learning-Logic. Invention Report S81-64, File 1. Office of Technology Licensing, Stanford University, Stanford, CA.

Patterson, K. (1990). Alexia and neural nets. Japanese Journal of Neuropsychology 6, pp. 90–99.

Patterson, K., Lambon Ralph, M. A. (1999). Selective disorders of reading? Current Opinion in Neurobiology 9, pp. 235–239.

Patterson, K., Morton, J. (1985a). From orthography to phonology: An attempt at an old interpretation. In: Patterson, K., Marshall, J. C., Coltheart, M. (Eds.), Surface dyslexia: Neurophysiological and cognitive studies of phonological reading. Lawrence Erlbaum Associates Ltd., Hove, UK, pp. 335–359.

Patterson, K., Morton, J. (1985b). From orthography to phonology: An attempt at an old interpretation. In: Patterson, K., Marshall, J. C., Coltheart, M. (Eds.), Surface dyslexia: Neurophysiological and cognitive studies of phonological reading. Lawrence Erlbaum Associates Ltd., Hove, UK, pp. 335–359.

Patterson, K., Seidenberg, M. S., McClelland, J. L. (1989). Connections and disconnections: Acquired dyslexia in a computational model of reading processes. In: Morris, R. G. M. (Ed.), Parallel Distributed Processing: Implications for Psychology and Neurobiology. Oxford University Press, Oxford, pp. 131–181.

Patterson, K. E., Marshall, J. C., Coltheart, M. (Eds.) (1985). Surface dyslexia: neuropsychological and cognitive studies of phonological reading. Lawrence Erlbaum Associates, Hillsdale, NJ.

Paulesu, E., Frith, C. D., Frackowiak, R. S. J. (1993). The neural correlates of the verbal component of working memory. Nature 362, pp. 342–345.

Paulesu, E., Frith, U., Snowling, M., Gallagher, A., Morton, J., Frackowiak, R. S., Frith, C. D. (1996). Is developmental dyslexia a disconnection syndrome? Evidence from PET scanning. Brain 119, pp. 143–57.

Paulesu, E., Goldacre, B., Scifo, P., Cappa, S. F., Gilardi, M. C., Castiglioni, I., Perani, D., Fazio, F. (1997). Functional heterogeneity of left inferior frontal cortex as revealed by fMRI. Neuroreport 8 (8), pp. 2011–7.

Paulesu, E., McCrory, E., Fazio, F., Menoncello, L., Brunswick, N., Cappa, S. F., Cotelli, M., Cossu, G., Corte, F., Lorusso, M., Pesenti, S., Gallagher, A., Perani, D., Price, C., Frith, C. D., Frith, U. (2000). A cultural effect on brain function. Nature Neuroscience 3 (1), pp. 91–96.

Pavlov, I. P. (1927). Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex. Oxford University Press.

Paz, A. (1971). Introduction to probabilistic automata. Academic Press.

Pearlmutter, B. A. (1989). Learning State Space Trajectories in Recurrent Neural Networks. In: International Joint Conference on Neural Networks, pp. 365–372.

Peeva, M. G., Guenther, F. H., Tourville, J. a., Nieto-Castanon, A., Anton, J.-L., Nazarian, B., Alario, F.-X. (2010). Distinct representations of phonemes, syllables, and supra-syllabic sequences in the speech production network. NeuroImage 50 (2), pp. 626–38.

Perani, D., Cappa, S. F., Schnur, T., Tettamanti, M., Collina, S., Rosa, M. M., Fazio, F. (1999). The neural correlates of verb and noun processing. A PET study. Brain 122 (Pt 1), pp. 2337–2344.

Pereira, F., Mitchell, T., Botvinick, M. (2009). Machine learning classifiers and fMRI: a tutorial overview. NeuroImage 45 (1 Suppl), pp. S199–209.

Perry, C., Ziegler, J. C., Zorzi, M. (2007). Nested incremental modeling in the development of computational theories: the CDP+ model of reading aloud. Psychological Review 114 (2), pp. 273–315.

Perry, C., Ziegler, J. C., Zorzi, M. (2010). Beyond single syllables: Large-scale modeling of reading aloud with the Connectionist Dual Process (CDP++) model. Cognitive Psychology 61, pp. 106–151.

Petersen, S. E., Fiez, J. A. (1993). The processing of single words studied with positron emission tomography. Annual Review of Neuroscience 16, pp. 509–530.

Pfeifer, R. (1994). The "Fungus Eater Approach" to Emotion: A View from Artificial Intelligence. Cognitive Studies: Bulletin of the JApanese Cognitive Science Society 1 (2), pp. 42–57.

Picard, R. W. (1997). Affective Computing. MIT Press.

Picard, R. W., Vyzas, E., Healey, J. (2001). Toward machine emotional intelligence: Analysis of affective physiological state. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (10), pp. 1175–1191.

Pineda, F. (1987). Generalization of back-propagation to recurrent neural networks. Physical review letters 59 (19), pp. 2229–2232.

Pioggia, G., Igliozzi, R., Ferro, M., Ahluwalia, A., Muratori, F., De Rossi, D. (2005). An android for enhancing social skills and emotion recognition in people with autism. IEEE Transactions on Neural Systems and Rehabilitation Engineering 13 (4), pp. 507–515.

Platt, J. (1991). A Resource-Allocating Network for Function Interpolation. Neural Computation 3 (2), pp. 213–225.

Plaut, D. C. (1997). Structure and Function in the Lexical System: Insights from Distributed Models of Word Reading and Lexical Decision. Language and Cognitive Processes 12 (5-6), pp. 765–806.

Plaut, D. C. (1999). A Connectionist Approach to Word Reading and Acquired Dyslexia: Extension to Sequential Processing. Cognitive Science 23 (4), pp. 543–568.

Plaut, D. C. (2005). Connectionist approaches to reading. In: Snowling, M. J., Hulme, C. (Eds.), The science of reading: A handbook. Blackwell, pp. 24–38.

Plaut, D. C., Kello, C. T. (1999). The emergence of phonology from the interplay of speech comprehension and production: A distributed connectionist approach. In: MacWhinney, B. (Ed.), The emergence of language. Lawrence Erlbaum Associates, Inc., pp. 381–416.

Plaut, D. C., McClelland, J. L. (2010). Locating object knowledge in the brain: comment on Bowers's (2009) attempt to revive the grandmother cell hypothesis. Psychological Review 117 (1), pp. 284–8.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., Patterson, K. (1996). Understanding normal and impaired word reading: computational principles in quasi-regular domains. Psychological Review 103 (1), pp. 56–115.

Plaut, D. C., Nowlan, S. J., Hinton, G. E. (1986). Experiments on learning by back propagation. Tech. Rep. CMU-CS-86-126, CS Dept., Carnegie Mellon University, Pittsburgh, PA.

Plaut, D. C., Shallice, T. (1993). Deep dyslexia: A case study of connectionist neuropsychology. Cognitive Neuropsychology 10 (5), pp. 377–500.

Plunkett, K., Marchman, V. (1993). From rote learning to system building: acquiring verb morphology in children and connectionist nets. Cognition 48 (1), pp. 21–69.

Poggio, T., Girosi, F. (1990). Regularization algorithms for learning that are equivalent to multilayer networks. Science 247 (4945), pp. 978–82.

Poldrack, R. A., Wagner, A. D., Prull, M. W., Desmond, J. E., Glover, G. H., Gabrieli, J. D. E. (1999). Functional Specialization for Semantic and Phonological Processing in the Left Inferior Prefrontal Cortex. NeuroImage 10, pp. 15–35.

Price, C. J. (2000). The anatomy of language: contributions from functional neuroimaging. Journal of Anatomy 197 (Pt 3), pp. 335–359.

Price, C. J., Devlin, J. T. (2003). The myth of the visual word form area. NeuroImage 19 (3), pp. 473–481.

Price, C. J., Gorno-Tempini, M. L., Graham, K. S., Biggio, N., Mechelli, A., Patterson, K., Noppeney, U. (2003). Normal and pathological reading: converging data from lesion and imaging studies. NeuroImage 20 Suppl 1, pp. S30–41.

Price, C. J., McCrory, E., Noppeney, U., Mechelli, A., Moore, C. J., Biggio, N., Devlin, J. T. (2006). How reading differs from object naming at the neuronal level. NeuroImage 29 (2), pp. 643–648.

Price, C. J., Mechelli, A. (2005). Reading and reading disturbance. Current Opinion in Neurobiology 15 (2), pp. 231–238.

Price, C. J., Moore, C. J., Frackowiak, R. S. (1996a). The effect of varying stimulus rate and duration on brain activity during reading. NeuroImage 3 (1), pp. 40–52.

Price, C. J., Wise, R. J., Frackowiak, R. S. (1996b). Demonstrating the implicit processing of visually presented words and pseudowords. Cerebral Cortex 6 (1), pp. 62–70.

Price, C. J., Wise, R. J., Watson, J. D., Patterson, K., Howard, D., Frackowiak, R. S. (1994). Brain activity during reading. The effects of exposure duration and task. Brain 117 (6), pp. 1255–1269.

Pugh, K. R., Mencl, W. E., Jenner, A. R., Katz, L., Frost, S. J., Lee, J. R., Shaywitz, S. E., Shaywitz, B. A. (2000). Functional neuroimaging studies of reading and reading disability (developmental dyslexia). Mental Retardation and Developmental Disabilities Research Reviews 6 (3), pp. 207–213.

Rabin, M. O. (1963). Probabilistic automata. Information and Control 6 (3), pp. 230–245.

Radaelli, M. (2010). Metodologie di clustering per analisi di dati di attivazione corticale, Unpublished Master's Thesis (in Italian), Università degli Studi di Milano.

Raichle, M. E. (1994). Images of the Mind: Studies with Modern Imaging Techniques. Annual Review of Psychology 45, pp. 333–356.

Raichle, M. E. (2003). Functional brain imaging and human brain function. The Journal of Neuroscience 23 (10), pp. 3959–62.

Rapcsak, S. Z., Beeson, P. M., Henry, M. L., Leyden, A., Kim, E., Rising, K., Andersen, S., Cho, H. (2009). Phonological dyslexia and dysgraphia: cognitive mechanisms and neural substrates. Cortex 45 (5), pp. 575–91.

Rastle, K., Coltheart, M. (1998). Whammy and double whammy: The effect of length on nonword reading. Psychonomic Bulletin and Review 5, pp. 277–282.

Rastle, K., Coltheart, M. (1999). Serial and strategic effects in reading aloud. Journal of Experimental Psychology: Human Perception and Performance 25, pp. 482–503.

Ratcliff, R. (1978). A theory of memory retrieval. Psychological Review 85 (2), pp. 59–108.

Ratcliff, R. (1980). A note on modeling accumulation of information when the rate of accumulation changes over time. Journal of Mathematical Psychology 21 (2), pp. 178–184.

Ratcliff, R. (1981). A theory of order relations in perceptual matching. Psychological Review 88 (6), pp. 552–572.

Ratcliff, R. (1988). Continuous Versus Discrete Information Processing: Modeling Accumulation of Partial Information. Psychological Review 95 (2), pp. 238–255.

Reilly, W. S., Bates, J. (1992). Building Emotional Agents. Tech. Rep. CMU-CS-92-143, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Rescorla, R. A., Wagner, A. R. (1972). A theory of pavlovian conditioning: The effectiveness of reinforcement and non-reinforcement. In: Black, A. H., Prokasy, W. F. (Eds.), Classical Conditioning II: Current Research and Theory. Appleton-Century-Crofts, New York, NY, pp. 64–69.

Ricks, D. J., Colton, M. B. (2010). Trends and considerations in robot-assisted autism therapy. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4354–4359.

Rohde, D. L. T. (1999). LENS: The light, efficient network simulator. Tech. Rep. CMU-CS-99-164, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, *http://tedlab.mit.edu:16080/dr/lens/*.

Rohde, D. L. T., Gonnerman, L. M., Plaut, D. C. (Submitted). An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence.

Rolls, E. T. (1999). The brain and emotion. Oxford University Press.

Rorden, C., Brett, M. (2000). Stereotaxic display of brain lesions. Behavioral Neurology 12 (4), pp. 191–200.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65 (6), pp. 386–408.

Rosenblatt, F. (1962). Principles of neurodynamics; perceptrons and the theory of brain mechanisms. Spartan Books.

Rumelhart, D. E., Durbin, R., Golden, R., Chauvin, Y. (1996). Backpropagation: the basic theory. In: Mathematical perspectives on neural networks. Erlbaum, pp. 533–566.

Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). Learning internal representations by error propagation. In: McClelland, J. L., Rumelhart, D. E., the PDP Research Group (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition. Vol. I. MIT Press, pp. 318–362.

Rumelhart, D. E., McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. the contextual enhancement effect and some tests and extensions of the model. Psychological Review 89 (1), pp. 60–94.

Rumelhart, D. E., McClelland, J. L. (1986). On learning the past tenses of english verbs. In: McClelland, J. L., Rumelhart, D. E., the PDP Research Group (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition. Vol. II. MIT Press, pp. 216–271.

Rumelhart, D. E., Zipser, D. (1985). Feature discovery by competitive learning. Cognitive Science 9, pp. 75–112.

Rumsey, J. M., Horwitz, B., Donohue, B. C., Nace, K., Maisog, J. M., Andreason, P. (1997). Phonological and orthographic components of word recognition. A PET-rCBF study. Brain 120 (Pt 5), pp. 739–759.

Ruspini, E. H. (1969). A new approach to clustering. Information and Control 15 (1), pp. 22–32.

Russell, J. A. (1980). A circumplex model of affect. Journal of Personality and Social Psychology 39, pp. 1161–1178.

Russell, S., Norvig, P. (1995). Artificial Intelligence: A modern approach. Prentice Hall.

Ryckman, R. M. (2003). Theories of Personality, 8th Edition. Wadsworth Publishing.

Sahdra, B., Thagard, P. (2003). Self-Deception and Emotional Coherence. Minds and Machines 13, pp. 213–231.

Salimi-Khorshidi, G., Smith, S. M., Keltner, J. R., Wager, T. D., Nichols, T. E. (2009). Meta-analysis of neuroimaging data: a comparison of image-based and coordinate-based pooling of studies. NeuroImage 45 (3), pp. 810–823.

Schachter, S., Singer, J. E. (1962). Cognitive, social, and physiological determinants of emotional state. Psychological Review 69, pp. 379–99.

Scherer, K. R. (1984a). Emotion as a multicomponent process: a model and some cross-cultural data. Review of Personality and Social Psychology 5, pp. 37–63.

Scherer, K. R. (1984b). On the nature and function of emotion: A component process approach. In: Scherer, K. R., Ekman, P. (Eds.), Approaches to emotion. Lawrence Erlbaum Associates, pp. 293–317.

Scherer, K. R. (2005). What are emotions? and how can they be measured? Social Science Information 44 (4), pp. 695–729.

Scherer, K. R., Schorr, A., Johnstone, T. (Eds.) (2001). Appraisal Processes in Emotion: Theory, Methods, Research. Oxford University Press.

Schlosberg, H. (1954). Three dimensions of emotion. Psychological Review 61 (2), pp. 81–88.

Schölkopf, B., Smola, A. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA.

Schölkopf, B., Smola, A., Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10 (5), pp. 1299–1319.

Schonbein, W., Bechtel, W. (2003). History of Cognitive Science and Computational Modeling. In: Nadel, L. (Ed.), Encyclopedia of cognitive science. Nature Publishing Group/Macmillan.

Schroeder, M. (2004). Speech and emotion research: an overview of research frameworks and a dimensional approach to emotional speech synthesis. Ph.D. thesis, Saarland University.

Schulte, J., Rosenberg, C., Thrun, S. (1999). Spontaneous, short-term interaction with mobile robots. In: IEEE International Conference on Robotics & Automation, pp. 658–663.

Scott, S. K., Young, A. W., Calder, A. J., Hellawell, D. J., Aggleton, J. P., Johnson, M. (1997). Impaired auditory recognition of fear and anger following bilateral amygdala lesions. Nature 385 (6613), pp. 254–257.

Searle, J. R. (1980). Minds, Brains, and Programs. The Behavioral and Brain Sciences 3, pp. 417–424.

Sebe, N., Cohen, I., Gevers, T., Huang, T. S. (2006). Emotion Recognition Based on Joint Visual and Audio Cues. In: Proceedings of the 18th International Conference on Pattern Recognition, pp. 1136–1139.

Seghier, M., Lee, H. L., Schofield, T., Ellis, C. L., Price, C. J. (2008). Inter-subject variability in the use of two different neuronal networks for reading aloud familiar words. NeuroImage 42 (3), pp. 1226–1236.

Seidenberg, M. S. (in press). Computational Models of Reading: Connectionist and Dual-Route Approaches. In: Spivey, M., McRae, K., Joanisse, M. (Eds.), Cambridge handbook of psycholinguistics. Cambridge University Press.

Seidenberg, M. S., McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. Psychological Review 96 (4), pp. 523–568.

Seidenberg, M. S., Waters, G. S., Barnes, M. A., Tanenhaus, M. K. (1984). When does irregular spelling or pronunciation influence word recognition? Journal of Verbal Learning and Verbal Behaviour 23, pp. 383–404.

Sejnowski, T. J., Rosenberg, C. R. (1987). Parallel networks that learn to pronounce english text. Complex Systems 1, pp. 145–168.

Sejonwski, T. J., Rosenberg, J. R. (1986). Nettalk: a parallel network that learns to read aloud. Tech. Rep. JHU/EECS-86/01, Johns Hopkins University.

Shallice, T., Vallar, G. (1990). The impairment of auditory-verbal short-term storage. In: Vallar, G., Shallice, T. (Eds.), Neuropsychological impairments of short-term memory. Cambridge University Press, Ch. 1, pp. 11–53.

Shimokawa, T., Sawaragi, T. (2001). Acquiring Communicative Motor Acts of Social Robot Using Interactive Evolutionary Computation. In: Proceedings of IEEE International Conference on Systems, Man & Cybernetics, pp. 1396–1401.

Si, M., Marsella, S. C., Pynadath, D. V. (2009). Modeling appraisal in theory of mind reasoning. Autonomous Agents and Multi-Agent Systems 20 (1), pp. 14–31.

Sibley, D., Kello, C., Seidenberg, M. (2010). Learning orthographic and phonological representations in models of monosyllabic and bisyllabic naming. European Journal of Cognitive Psychology.

Sibley, D. E., Kello, C. T. (2005). A computational exploration of double dissociations: modes of processing instead of components of processing. Cognitive Systems Research 6 (1), pp. 61–69.

Sibley, D. E., Kello, C. T., Plaut, D. C., Elman, J. L. (2008). Large-scale modeling of wordform learning and representation. Cognitive science 32 (4), pp. 741–754.

Sibson, R. (1972). Order invariant methods for data analysis (with discussion). Journal of the Royal Statistical Society B 34, pp. 311–349.

Siegle, G. J., Ingram, R. E. (1997). Modeling individual differences in negative information processing biases. In: Matthews, G. (Ed.), Cognitive Science Perspectives on Personality and Emotion. Elsevier, Ch. 7, pp. 301–353.

Simon, H. A. (1967). Motivational and emotional controls of cognition. Psychological Review 74 (1), pp. 29–39.

Sobol-Shikler, T., Robinson, P. (2010). Classification of complex information: inference of co-occurring affective states from their expressions in speech. IEEE transactions on Pattern Analysis and Machine Intelligence 32 (7), pp. 1284–97.

Spieler, D. H., Balota, D. A. (1997). Bringing computational models of word naming down to the item level. Psychological Science 8 (6), pp. 411–416.

St. John, M. F., McClelland, J. L. (1990). Learning and Applying Contextual Constraints in Sentence Comprehension. Artificial Intelligence 46, pp. 217–257.

Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M. (2006). BioGRID: a general repository for interaction datasets. Nucleic Acids Res. 34, pp. D535–D539.

Steinbach, M., Karypis, G., Kumar, V. (2000). A comparison of document clustering techniques. Tech. Rep. 00-034, University of Minnesota.

Strain, E., Patterson, K., Seidenberg, M. S. (1995). Semantic effects in single-word naming. Journal of Experimental Psychology: Learning, Memory, and Cognition 21 (5), pp. 1140–1154.

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. Journal of Experimental Psychology 18 (1), pp. 643–662.

Sun, R. (Ed.) (2008a). Cambridge handbook of computational psychology. Cambridge University Press.

Sun, R. (2008b). Introduction to computational cognitive modeling. In: Sun, R. (Ed.), Cambridge handbook of computational psychology. Cambridge University Press, pp. 3–19.

Sun, R., Coward, L. A., Zenzen, M. J. (2005). On levels of cognitive modeling. Philosophical Psychology 18 (5), pp. 613–637.

Sutton, R. S., Barto, A. G. (1998). Reinforcement Learning: An Introduction. The MIT Press, Cambridge, MA.

Talairach, J., Tournoux, P. (1988). Co-planar stereotaxic atlas of the human brain: 3-D proportional system: An approach to cerebral imaging. Thieme Medical Publishers, New York, NY.

Tan, L. H., Laird, A. R., Li, K., Fox, P. T. (2005). Neuroanatomical Correlates of Phonological Processing of Chinese Characters and Alphabetic Words: A Meta-Analysis. Human Brain Mapping 25 (1), pp. 83–91.

Taraban, R., McClelland, J. L. (1987). Conspiracy effects in word pronunciation. Journal of Memory and Language 26, pp. 608–631.

Thagard, P. (2003). Why wasn't O.J. convicted? Emotional coherence in legal inference. Cognition & Emotion 17 (3), pp. 361–383.

Thagard, P., Nerb, J. (2002). Emotional Gestalts: Appraisal, Change, and the Dynamics of Affect. Personality and Social Psychology Review 6 (4), pp. 274–282.

The Gene Ontology Consortium (2000). Gene ontology: tool for the unification of biology. Nature Genetics 25, pp. 25–29.

Thomas, M. S. C., McClelland, J. L. (2008). Connectionist models of cognition. In: Sun, R. (Ed.), Cambridge handbook of computational psychology. Cambridge University Press, pp. 23–59.

Tian, Y.-I., Kanade, T., Cohn, J. (2001). Recognizing Action Units for facial expression analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2), pp. 97–115.

Treiman, R., Kessler, B., Bick, S. (2003). Influence of consonantal context on the pronunciation of vowels: A comparison of human readers and computational models. Cognition 88, pp. 49–78.

Turing, A. M. (1936). On computable numbers with an application to the entscheidungs problem. Proceedings of the London Mathematical Society 2, pp. 230–265.

Turk, M., Pentland, A. (1991). Eigenfaces for Recognition. Journal of Cognitive Neuroscience 3 (1), pp. 71–86.

Turkeltaub, P. (2002). Meta-Analysis of the Functional Neuroanatomy of Single-Word Reading: Method and Validation. NeuroImage 16 (3), pp. 765–780.

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., Mazoyer, B., Joliot, M. (2002). Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. NeuroImage 15 (1), pp. 273–89.

Umiltà, C. (Ed.) (1999). Manuale di Neuroscienze, 2nd Edition. Il Mulino.

Valdois, S., Carbonnel, S., Juphard, A., Baciu, M., Ans, B., Peyrin, C., Segebarth, C. (2006). Polysyllabic pseudo-word processing in reading and lexical decision: converging evidence from behavioral data, connectionist simulations and functional MRI. Brain Research 1085 (1), pp. 149–162.

Vallar, G., Di Betta, a. M., Silveri, M. C. (1997). The phonological short-term store-rehearsal system: patterns of impairment and neural correlates. Neuropsychologia 35 (6), pp. 795–812.

Valstar, M. F., Gunes, H., Pantic, M. (2007). How to distinguish posed from spontaneous smiles using geometric features. In: Proceedings of the 9th International Conference on Multimodal Interfaces, pp. 38–45.

Van der Kloot, W. A., Spaans, A. M. J., Heiser, W. J. (2005). Instability of Hierarchical Cluster Analysis Due to Input Order of the Data: The PermuCLUSTER Solution. Psychological Methods 10 (4), pp. 468–476.

Vanier, M., Caplan, D. (1985). CT Scan Correlates of Surface Dyslexia. In: Patterson, K. E., Marshall, J. C., Coltheart, M. (Eds.), Surface dyslexia: Neuropsychological and cognitive studies of phonological reading. Lawrence Erlbaum Associates, Hillsdale, NJ, Ch. 18, pp. 511–522.

Velásquez, J. D. (1997). Modeling emotions and other motivations in synthetic agents. In: Proceedings of the Fourteenth National Conference on Artificial Intelligence, pp. 10–15.

Ververidis, D., Kotropoulos, C. (2006). Emotional speech recognition: Resources, features, and methods. Speech Communication 48 (9), pp. 1162–1181.

Vigneau, M., Jobard, G., Mazoyer, B., Tzourio-Mazoyer, N. (2005). Word and non-word reading: what role for the Visual Word Form Area? NeuroImage 27 (3), pp. 694–705.

Vinckier, F., Dehaene, S., Jobert, A., Dubus, J. P., Sigman, M., Cohen, L. (2007). Hierarchical coding of letter strings in the ventral stream: dissecting the inner organization of the visual word-form system. Neuron 55 (1), pp. 143–156.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. Kybernetik 14, pp. 85–100.

von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and Computing 17 (4), pp. 395–416.

Von Luxburg, U., Ben-David, S. (2005). Towards a Statistical Theory of Clustering. In: Proceedings of the PASCAL workshop on Statistics and Optimization of Clustering.

Wager, T. D., Barrett, L. F., Bliss-Moreau, E., Lindquist, K. A., Duncan, S., Kober, H., Joseph, J., Davidson, M., Mize, J. (2008). The neuroimaging of emotion. In: Lewis, M., Haviland-Jones, J. M., Barrett, L. F. (Eds.), Handbook of Emotion. The Guildford Press.

Wager, T. D., Lindquist, M., Kaplan, L. (2007). Meta-analysis of functional neuroimaging data: current and future directions. Social Cognitive and Affective Neuroscience 2 (2), pp. 150–158.

Wager, T. D., Lindquist, M. A., Nichols, T. E., Kober, H., Van Snellenberg, J. X. (2009). Evaluating the consistency and specificity of neuroimaging data using meta-analysis. NeuroImage 45 (1 Suppl), pp. S210–21.

Wager, T. D., Phan, K. L., Liberzon, I., Taylor, S. F. (2003). Valence, gender, and lateralization of functional brain anatomy in emotion: a meta-analysis of findings from neuroimaging. Neuroimage 19 (3), pp. 513–531.

Wager, T. D., Smith, E. E. (2003). Neuroimaging studies of working memory: a meta-analysis. Cognitive, Affective & Behavioral Neuroscience 3 (4), pp. 255–274.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58, pp. 236–244.

Watkins, C. (1989). Learning from delayed rewards. Ph.D. thesis, University of Cambridge, UK.

Weekes, B. S. (1997). Differential effects of number of letters on word and nonword naming latency. Quarterly Journal of Experimental Psychology 50A, pp. 439–456.

Wells, A., Matthews, G. (1994). Attention and emotion: a clinical perspective. Lawrence Erlbaum Associates.

Wells, A., Matthews, G. (1996). Modelling cognition in emotional disorder: the s-ref model. Behavioral Research Therapy 34 (11-12), pp. 881–888.

Wernicke, C. (1874). Der aphasische Symptomencomplex: Eine psychologische Studie auf anatomischer Basis. Max Cohn & Weigart, Breslau.

Whalen, P. J., Rauch, S. L., Etcoff, N. L., McInerney, S. C., Lee, M. B., Jenike, M. A. (1998). Masked presentations of emotional facial expressions modulate amygdala activity without explicit knowledge. The Journal of Neuroscience 18 (1), pp. 411–418.

Widrow, B., Hoff, M. E. (1960). Adaptive switching circuits. IRE WESCON Convention Record, pp. 96–104.

Williams, R. J., Zipser, D. (1989). Learning algorithm for continually running fully recurrent neural networks. Neural Computation 1, pp. 270–280.

Williams, R. J., Zipser, D. (1995). Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity. In: Chauvin, Y., Rumelhart, D. E. (Eds.), Back propagation: Theory, Architectures and Applications. Erlbaum, Hillsdale, NJ, pp. 433–486.

Wright, I., Sloman, A. (1997). Minder1: An implementation of a protoemotional agent architecture. Tech. Rep. CSRP-97-1, University of Birmingham, School of Computer Science.

Wright, I., Sloman, A., Beaudoin, L. (1996). Towards a design-based analysis of emotional episodes. Philosophy Psychiatry and Psychology 3, pp. 101–126.

Xu, R., Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks 16 (3), pp. 645–78.

Xu, R., Wunsch, D. (2008). Clustering. IEEE Press Series on Computational Intelligence. IEEE Press – Wiley.

Yacoob, Y., Davis, L. S. (1996). Recognizing human facial expressions from long image sequences using optical flow. IEEE Transaction on Pattern Analysis and Machine Intelligence 18 (6), pp. 636–642.

Yager, R., Filev, D. (1994). Approximate clustering via the mountain method. IEEE Transactions on Systems, Man, and Cybernetics 24 (8), pp. 1279–1284.

Yates, M. (2005). Phonological neighbors speed visual word processing: evidence from multiple tasks. Journal of Experimental Psychology: Learning, Memory, and Cognition 31, pp. 1385–1397.

Zadeh, L. A. (1965). Fuzzy sets. Information and Control 8 (3), pp. 338–353.

Zanna, M., Rempel, J. K. (1988). Attitudes: A new look at an old concept. In: Bar-Tal, D., Kruglanski, A. W. (Eds.), The social psychology of knowledge. Cambridge University Press, Cambridge, UK, pp. 315–334.

Zeki, S., Watson, J. D., Lueck, C. J., Friston, K. J., Kennard, C., Frackowiak, R. S. J. (1991). A direct demonstration of functional specialization in human visual cortex. Journal of Neuroscience 11, pp. 641–649.

Zeng, Z., Hu, Y., Liu, M., Fu, Y., Huang, T. S. (2006). Training combination strategy of multi-stream fused hidden Markov model for audio-visual affect recognition. In: Proceedings of the 14th annual ACM international conference on Multimedia, pp. 65–68.

Zeng, Z., Pantic, M., Roisman, G. I., Huang, T. S. (2009). A survey of affect recognition methods: audio, visual, and spontaneous expressions. IEEE transactions on pattern analysis and machine intelligence 31 (1), pp. 39–58.

Zhang, T., Ramakrishnan, R., Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. ACM SIGMOD Record 25 (2), pp. 103–114.

Zorzi, M. (2005). Computational models of reading. In: Houghton, G. (Ed.), Connectionist models in psychology. Psychology Press, pp. 403–444.

Zorzi, M., Houghton, G., Butterworth, B. (1998). Two Routes or One in Reading Aloud? A Connectionist Dual-Process Model. Journal of Experimental Psychology: Human Perception and Performance 24 (4), pp. 1131–1161.

List of Figures

List of Tables