Neural Networks 166 (2023) 634–644

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Solving classification tasks by a receptron based on nonlinear optical speckle fields

B. Paroli, G. Martini, M.A.C. Potenza, M. Siano, M. Mirigliano, P. Milani *

CIMAINA and Dipartimento di Fisica, Università degli Studi di Milano, via G. Celoria 16, 20133, Milan, Italy

ARTICLE INFO

Article history:
Received 30 September 2022
Received in revised form 7 June 2023
Accepted 2 August 2023
Available online 9 August 2023

Keywords:
Perceptron
Classification
Non-linear networks
Optical device
Boolean functions

ABSTRACT

Among several approaches to tackle the problem of energy consumption in modern computing systems, two solutions are currently investigated: one consists of artificial neural networks (ANNs) based on photonic technologies, the other is a different paradigm compared to ANNs and it is based on random networks of non-linear nanoscale junctions resulting from the assembling of nanoparticles or nanowires as substrates for neuromorphic computing. These networks show the presence of emergent complexity and collective phenomena in analogy with biological neural networks characterized by self-organization, redundancy, and non-linearity. Starting from this background, we propose and formalize a generalization of the perceptron model to describe a classification device based on a network of interacting units where the input weights are non-linearly dependent. We show that this model, called "receptron", provides substantial advantages compared to the perceptron as, for example, the solution of non-linearly separable Boolean functions with a single device. The receptron model is used as a starting point for the implementation of an all-optical device that exploits the non-linearity of optical speckle fields produced by a solid scatterer. By encoding these speckle fields we generated a large variety of target Boolean functions. We demonstrate that by properly setting the model parameters, different classes of functions with different multiplicity can be solved efficiently. The optical implementation of the receptron scheme opens the way for the fabrication of a completely new class of optical devices for neuromorphic data processing based on a very simple hardware.

1. Introduction

The current prominent approach to neuromorphic computing is based on the notion that the modification of synaptic connection strengths in the brain allows to learn and to perform complex tasks (Ambrogio et al., 2018; Lillicrap et al., 2020; Lynn & Bassett, 2019; Pershin & Di Ventra, 2010; Rajendran, Sebastian, Schmuker, Srinivasa, & Eleftheriou, 2019). The possibility of adjusting independently the synaptic weights to respond to different input patterns has been translated in the perceptron learning rule where a biological or artificial neuron is modeled as a simple linear summation and thresholding device (Jeong, Kim, Ziegler, & Kohlstedt, 2013; McCulloch & Pitt, 1943; Nagy, 1991; Rosenblatt, 1958). Ensembles of connected perceptrons constitute artificial neural networks (ANNs) where the weight of each unit can be independently modified, accordingly to a learning rule, to obtain a target output, given a certain input (Liu, Yu, & Chai, 2021; Schuman et al., 2022; Xia & Yang, 2019). The use of crossbar memristive arrays has been proposed in order to reduce the energy payload of ANNs: in this approach all nodes are linearly independent and the equivalent of the corresponding synaptic weight must be updated avoiding cross-talking between individual synapse nodes (Burr et al., 2016; Rajendran et al., 2019; Xia & Yang, 2019).

ANNs based on photonic technologies developed for telecom applications (Nakajima, Tanaka, & Hashimoto, 2021; Tomson et al., 2016) can represent a valid alternative to conventional electronic hardware for the achievement of a significant reduction of the operational power and increase of the speed and parallelism (Shen et al., 2017). Various photonic ANN models have been reported (Nakajima et al., 2021; Shen et al., 2017; Vandoorne et al., 2014): as in the case of their electronic counterpart, the architecture of optical ANNs is characterized by a lack of similarity with respect to biological neural systems, where self-organization, redundancy, non-linearity, and non-locality governs both structure and functions (Diaz-Alvarez et al., 2019; Hochstetter et al., 2021; Milano, Miranda, & Ricciardi, 2022). Neurons utilize a wealth of non-linear mechanisms to transform synaptic input into output firing (Ting-Ho Lo, 2011; Tononi, Boly,

* Corresponding author.
E-mail addresses: bruno.paroli@unimi.it (B. Paroli), gianluca.martini@unimi.it (G. Martini), marco.potenza@unimi.it (M.A.C. Potenza), mirko.siano@unimi.it (M. Siano), matteo.mirigliano@unimi.it (M. Mirigliano), paolo.milani@mi.infn.it (P. Milani).

https://doi.org/10.1016/j.neunet.2023.08.001

Massimini, & Koch, 2016); the majority of inputs to a neuron is received primarily through synapses made onto elaborate tree-like structures called dendrites (Häusser, Spruston, & Stuart, 2000), allowing transformations that extend far beyond the simple sum-and-threshold operation (Bicknell & Häusser, 2021). The morphology and the electrical properties of dendrites define the input–output relationship of neurons and the rules for the induction of synaptic plasticity (Silver, 2010).

A radical alternative to the top-down fabrication of electronic or optical ANNs is based on the use of networks consisting of a large number of non-linear nanoscale junctions resulting from the random assembling of nanoparticles or nanowires (Li et al., 2020). These networks show the presence of emergent complexity and collective phenomena in analogy with biological neural networks and, in particular, hierarchical collective dynamics (Mallinson et al., 2019), and heterosynaptic plasticity (Milano et al., 2020).

Networks of interconnected nanojunctions are characterized by the non-linear and distributed nature of the junction weight interactions: the weights are not univocally related to a single node since the highly interconnected junctions regulate their connectivity and the topology of conducting pathways depending on the input stimuli (Martini, Mirigliano, Paroli, & Milani, 2022; Mirigliano et al., 2021, 2020), in analogy to what observed in neuronal dendrites (Bicknell & Häusser, 2021). This aspect has been somehow neglected up to now, not considering that a consequence is the impossibility of applying the perceptron model to describe the evolution of the synaptic weights upon the interaction with external stimuli (Poirazi & Mel, 2001).

Recently we have shown that nanostructured Au films fabricated by assembling gold clusters, produced in the gas phase, have complex non-linear electrical properties and resistive switching behavior (Martini et al., 2022; Mirigliano et al., 2020; Mirigliano & Milani, 2021; Mirigliano et al., 2021). By interconnecting a generic pattern of electrodes with a cluster-assembled Au film, we demonstrated the fabrication of a device that can perform the binary classification of input signals, following a thresholding process, to generate a set of Boolean functions (Mirigliano et al., 2021). Considering the non-linear conduction properties of cluster-assembled gold films and their non-local response to input signals we underlined the inadequacy of a perceptron model with linearly independent weights proposing a model called "receptron" where the weights are not just associated with each input, but with their combinations (Martini et al., 2022; Mirigliano et al., 2021).

Here, we provide a general formalization of the receptron showing the fundamental differences between a receptron and a perceptron and we demonstrate that the general receptron model is not confined to electrical networks but it can be used as a starting point for the implementation of an all-optical device performing classification. We use a training approach based on a search procedure. The search procedure is independent from the complexity of the solved function and can be known a priori, allowing for its optimization. Therefore, the optimization of the search procedure can be performed at the maximum efficiency in terms of time-consuming for any possible function. An experimental demonstration of the use of an optical receptron for classification tasks is reported.

2. The receptron model

Starting from the traditional perceptron model (Minsky & Papert, 1970) based on linearly independent weights:

$$S = \sum_{j=1}^{n} x_j w_j^P , \tag{1}$$

where j numbers the inputs ($j \in [1, n]$) and w_j^P are constant real values, we formally introduce a more general form of Eq. (1), which allows for the non linear interaction of the inputs,

$$S = \sum_{j=1}^{n} x_j \tilde{w}_j(\vec{x}) \mid S \in R, \tag{2}$$

where $\tilde{w}_j(\vec{x}) : R^n \rightarrow C$ are complex-valued functions and $\vec{x} = (x_1, \ldots, x_n)$ is the input vector.

Eq. (2) is the basis of the receptron model: while Eq. (1) is a linear combination of the inputs (the weights are constant), in the receptron a modification of the inputs leads, in general, to a variation of the weights value $\tilde{w}_j(\vec{x})$, making the system extremely complex and allowing for the solution of problems not solvable through the simpler rules of a linear system.

As in the perceptron case the summation in Eq. (2) origins the activation of the receptron output through the thresholding process,

$$Y(x_1, \ldots, x_n) = \begin{cases} 1 & S > th \\ 0 & S \leq th \end{cases}, \tag{3}$$

where th is a constant threshold parameter. Eq. (3) can be written by using the Heaviside function as

$$Y(x_1, \ldots, x_n) = \Theta\left(S'\right), \tag{4}$$

where $S' = b + S$ and b is a constant bias.

With the aim of investigating the non-linear and statistical properties as well as the computing performance of the proposed model, we limit the model to purely Boolean inputs ($x_j \in \{0, 1\}$), this facilitates the analysis and helps to highlight the most important features. In this case, the weight functions $\tilde{w}_j(\vec{x})$ can be written with a finite number of parameters $w_{j_1 \ldots j_n}$, simplifying the model representation. In particular, we can Taylor-expand $\tilde{w}_j(\vec{x})$ and use the idempotency of Boolean variables $(x_j)^q = x_j \forall q \geq 1$ such that $S' = b + \sum_{j=1}^{n} x_j \tilde{w}_j(\vec{x})$ can be written as

$$S'(\vec{x}) = b + \sum_j w_j x_j + \sum_{j<k} w_{jk} x_j x_k + \sum_{j<k<l} w_{jkl} x_j x_k x_l + \cdots , \tag{5}$$

where $w_{j_1 \ldots j_n}$ are independent parameters that can be seen as the components of a tensor W ("weight tensor") of rank n and type $(n, 0)$. In Eq. (5) we have contracted, for simplicity of notation, the indexes of the matrix elements $w_{j_1 \ldots j_n}$ to a single index when all the indexes at the right are equals, as shown here below for $n = 3$,

$$\begin{aligned} w_j &\implies j = k = l \\ w_{jk} &\implies k = l \end{aligned} \tag{6}$$

or more in general

$$w_{j_1 \ldots j_m} \implies j_m = j_{m+1} = \cdots = j_n . \tag{7}$$

The expression in Eq. (5) is the most general Boolean-input function. In fact, a function defined over a finite set, such as the case of S', is completely specified by the values assumed over all the elements inside the domain. In the case of Boolean inputs, the domain contains 2^n elements: tuning every one of them allows to define the value of the function for each input. The number of coefficients in Eq. (5) is 1 for the bias, n for the diagonal elements w_j, $\frac{n(n-1)}{2}$ for w_{jk} and finally $\frac{n!}{m!(n-m)!}$ for the mth degree term, thus the total number is given by

$$\sum_{m=0}^{n} \binom{n}{m} = 2^n , \tag{8}$$

where we used the binomial theorem.

We have thus proved that the total number of independent parameters equals the number of input combinations, i.e., we

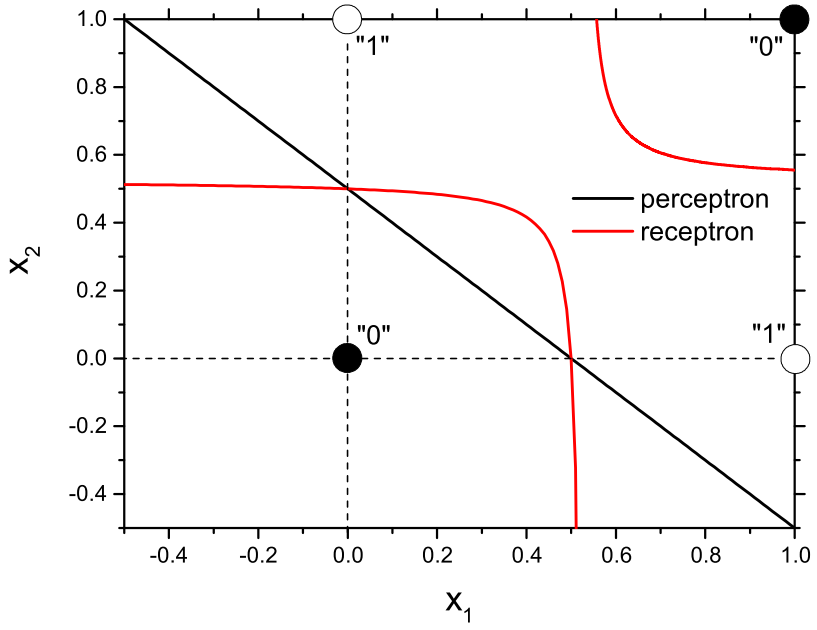

Fig. 1. Classification of the XOR output states with a single element. The black straight line and the red hyperbole are the decision boundaries for the perceptron and a receptron respectively, obtained by posing $S'(\vec{x}) = 0$ in Eq. (9). The high states (white circles) are separable from the low states (black circles) with the red hyperbole obtained with the receptron parameters $w_{11} = 1$, $w_{22} = 1$, $w_{12} = -1.9$ and $b = -0.5$ but, as is well known, they cannot be separated with the straight line generated by a single perceptron.

have different degrees of freedom for each input, that can be adjusted to implement any desired output.

The sum in Eq. (5) reduces to the perceptron case when off-diagonal terms of W vanish. As an example we consider $n = 2$, where we write the sum (5) to

$$S'(\vec{x}) = b + x_1 w_{11} + x_2 w_{22} + x_1 x_2 w_{12}. \tag{9}$$

In the perceptron case, the vanishing of w_{12} implies linearity, $S(1, 1) = S(0, 1) + S(1, 0)$. On the contrary, the off-diagonal elements yield $S(1, 1) \neq S(0, 1)+S(1, 0)$ for a receptron, meaning that the superposition principle is no longer valid, the latter terms being precisely responsible of the more complex non-linear interaction between the inputs.

Due to the missing coefficients, the number of independent parameters for a perceptron grows proportionally to the number of inputs, while the growth is exponential in a receptron (see Eq. (8)), highlighting its higher complexity. Therefore, classification of non-linearly separable functions can be realized with a single element, as shown in Fig. 1.

Here the output states of the XOR function are shown with black (low state) and white (high state) circles. The black straight line calculated with Eq. (9) by posing $S'(x) = 0$ and $w_{12} = 0$ cannot separate the black and white circles, therefore the XOR function cannot be implemented with a perceptron, contrarily the non-zero off-diagonal elements transform the straight line in the red hyperbole, separating the black and white circles, and making the XOR easy to classify.

Considering the generalization of the perceptron, we coined the term "receptron" (reservoir perceptron). Similarly to the case of reservoir computing, the variability in our model is generate by exploiting a static reservoir (a random assembly of electrically connected nanojunctions or the intensities of the optical speckle field (see below)). The reservoir computing exploits a high-dimensional dynamical system to convert the injected input into the trained output function both depending on time (Lukoševičius & Jaeger, 2009). The output function is obtained by means of a final layer, where the weight should be adjusted to obtain the useful function. In our approach, we use the high-dimensional behavior of a reservoir without exploiting its temporal evolution.

2.1. Optical receptron

In this section, we use the model formalized above to describe an optical receptron exploiting the interference of a large number of uncorrelated point-like sources, generating the non-linear interaction between optical inputs discussed above. The development of the optical model allows the explicit analytical calculation of the weights, useful for predicting the overall behavior of the device in terms of operation and degree of non-linearity.

Let us consider a set of n sources s_j positioned in a plane as sketched in Fig. 2. Since sources are uncorrelated, each will produce an electric field E_j with amplitudes A_j and phases ϕ_j at a given observation point P positioned at a distance d from the source plane

$$E_j = A_j e^{i\phi_j}\,. \tag{10}$$

Let us assume now that the light emission of each group of sources can be turned ON or OFF. This action can be formalized in our model, multiplying the field amplitudes in Eq. (10) with input Boolean variables $x_j \in \{0, 1\}$ as

$$E_j = x_j A_j e^{i\phi_j}. \tag{11}$$

The overall intensity in P will be given by

$$I = \left| \sum_{j=1}^{n} x_j A_j e^{i\phi_j} \right|^2 . \tag{12}$$

If we interpret the intensity at a point on the observation plane as the interaction between inputs (the input being the on/off pattern of sources) we see that the squared modulus in Eq. (12) allows for the non-linearity. We can calculate the weights of the optical receptron to show that this, rather than the perceptron,

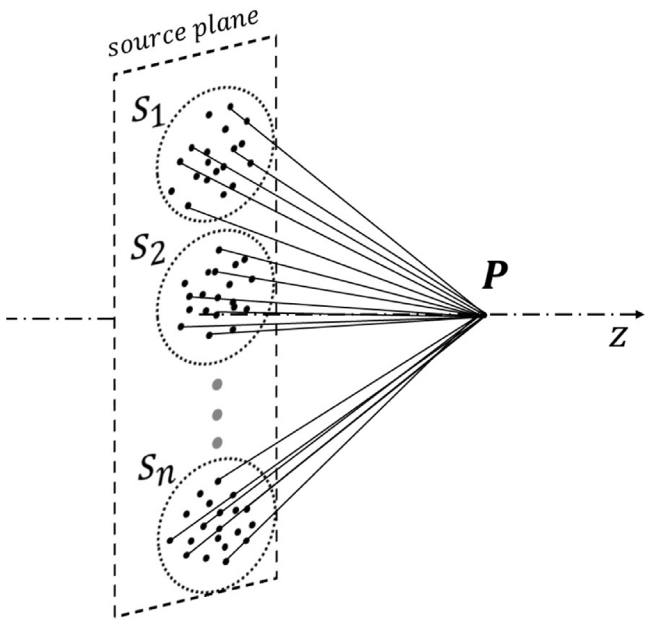

Fig. 2. Sketch of the optical implementation of a receptron exploiting the interference to create the non-linear interaction $I(\vec{x}) = S(\vec{x})$ of Eq. (13). The group of point-like sources s_j are independently on/off modulated through the Boolean inputs x_j. The intensity is observed at the point P along the optical axis z.

fits best such a device:

$$I = \sum_{j=1}^{n} x_j^2 A_j^2 + \sum_{j\neq k} x_j x_k A_j A_k e^{i(\phi_j-\phi_k)} = \\ = \sum_{j=1}^{n} x_j A_j^2 + \sum_{j<k} x_j x_k A_j A_k e^{i(\phi_j-\phi_k)} + \sum_{j<k} x_j x_k A_j A_k e^{i(\phi_k-\phi_j)} = \\ = \sum_{j=1}^{n} x_j A_j^2 + 2\cdot\sum_{j<k} x_j x_k A_j A_k cos(\phi_j-\phi_k)\,. \tag{13}$$

Eq. (13) shows that the intensity produced by the interference of groups of sources as a function of the input Boolean variables is consistent with the summation $S(\vec{x})$ of the proposed receptron model in Eq. (2).

Hence we pose $I(\vec{x}) = S(\vec{x})$ to find the weight functions of the optical receptron as

$$\tilde{w}_j(\vec{x}) = x_j A_j^2 + \sum_{\substack{k=1\\k\neq j}}^{N} x_k A_j A_k e^{i(\phi_j-\phi_k)}\,. \tag{14}$$

Combining Eqs. (13) and (5) we find the components of the weight tensor,

$$w_j = A_j^2 \tag{15}$$

$$w_{jk} = 2A_j A_k cos(\phi_j-\phi_k) \tag{16}$$

$$w_{j_1\cdot j_n} = 0 \quad elsewhere, \tag{17}$$

which demonstrate that the system can be fully described with the proposed model. The large number of sources used in the model is essential to impart high variability of the intensity in the space (depending on many independent parameters): a limited number of sources would still generate non-linearity of the intensity as a function of the inputs x_j, but the limited number of free parameters A_j, ϕ_j would produce a more regular field. Our idea is to maximize the variability of the intensities A_j and phases ϕ_j in the different observation points through a speckle field (Dainty, 1975; Goodman, 2007) generated with a random scatterer: this will in turn imply a higher spatial variability of the function $I(\vec{x}) = S(\vec{x})$ implemented by the network. The state of each node in the receptron does not evolve recursively in time as in a conventional reservoir (Rafayelyan, Dong, Tan, Krzakala, & Gigan, 2020) but only as a function of the input vector $\vec{x} = (x_1, \ldots, x_n)$, regardless of whether inputs vary over time or not. This is essential because in a receptron the output of the Boolean function must depend only on the input combination, both during the training process and during the data processing. Furthermore, with respect to a typical reservoir a receptron does not make use of a final layer to convert the reservoir output into the target output function (Mirigliano et al., 2021), but it uses a single threshold process applied to the analog outputs (intensity) of the reservoir, like in a perceptron, to generate a binary classification.

The group of sources described above, is experimentally realized with a scatterer illuminated by means of independent laser beams, as discussed in the next section.

3. Experimental setup

The experimental setup was realized for a 4-input device as shown in Fig. 3(a). Here a 5 mW He–Ne laser source was spatially filtered, collimated and split in four independent beams by means of a square optical-glass pyramid (the base of the pyramid is perpendicular to the optical axis). Each beam was sent to a mask comprising four optical apertures which physically implement the x_j of Eq. (11): all 16 (2^n) input combinations can be realized opening and closing the apertures.

The beams at the exit of the apertures impinge on an optical polypropylene scatterer ($\approx$ 100 μm thick) at 75 mm from the pyramid, in order to create four independent groups of uncorrelated point like sources as described in Section 2.1. The scatterer generates a homodyne speckle field that is detected with a Charge Coupled Device (CCD) on an observation plane, orthogonal to the propagation axis of the beams, at a distance of 125 mm from the scatterer. This distance is chosen in such a way the $\approx$ 40 μm speckle size (FWHM) is slightly larger than the CCD pixel size (7.4 μm). The speckle field is acquired in a 8 bit grey scale. The activation of each pixel, corresponding to a single receptron output, was implemented computationally by the thresholding operation (as shown in Eq. (3)), that converts the grey scale speckle pattern in a black and white Boolean field (see Fig. 3(b)). The high-resolution of the CCD (1600 $\times$ 1200 pixels) simultaneously detects a multitude of receptrons (with common inputs) in a completely parallel fashion.

In the current setup the stability has been tested over tens of minutes, in view of operation for prolonged time periods, two strategies can be applied in order to reduce the effects of system instabilities and of the noise: (i) a periodic training that should be realized in a characteristic period $\tau_s = \tau_t + \tau_c$, where τ_t is the training time and τ_c is the computing time, with $\tau_t \ll \tau_c$; (ii) a selection (during the training process) of the pixel solving the target function and having intensity (for all possible combinations) sufficiently far from the threshold of a noise margin D_N. In this way, the system becomes intrinsically immune to the noise or external disturbances, when they induce intensity variations less than D_N.

4. Results and discussion

4.1. Uniform speckle fields

In Fig. 4 we show an example of a uniform speckle field obtained with two different input combinations: (0, 0, 0, 1) and

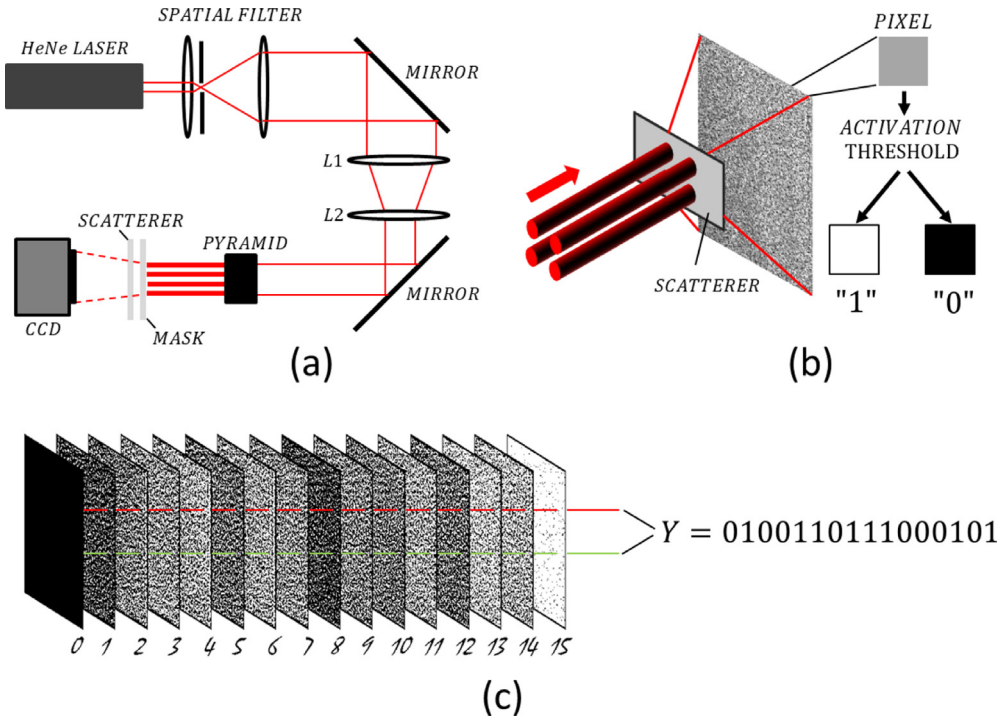

Fig. 3. (a) Experimental setup for the optical implementation of the receptron. A He–Ne laser beam is spatially filtered and collimated. The positive lens L1 and L2 are used to match the beam size to the optical pyramid base. The pyramid splits the main beam in four independent beams (the receptron inputs) impinging on the scatterer, after passing through a mask that modulates the beams ON/OFF. (b) Sketch of the method used to convert the 8 bit grey scale image of the speckle field in a corresponding black and white Boolean field. The speckle field, shown in grey scale, is perpendicular to the propagation axis of the laser beams (inputs) and generated with the optical pyramid (not shown). The scatterer plane is parallel to the CCD plane. Binarization is realized computationally for all the CCD pixels through a thresholding operation that converts each grey pixel in a black or white pixel as described in Eq. (3). (c) Example of solution of two identical Boolean functions. A stack of Boolean fields is obtained by changing the 16 input combinations. The two pixels shown in green and red, along the stack, generate the same output sequence $Y = 0100110111000101$ as a function of the inputs from 0 to 15.

$(1, 1, 1, 1)$ respectively. Since the average intensity of the raw images of the speckle fields across the CCD is not uniform due to the different position of the lasers on the optical scatterer, it is useful, for statistical analysis, to make uniform the entire image.

Therefore, the uniform speckle fields were numerically obtained by using the ratio $I(l, m) = I(l, m)_{CCD}/I(l, m)_{LPF}$, where $I(l, m)_{CCD}$ is the intensity of the speckle field acquired with the CCD, $I(l, m)_{LPF}$ is the same intensity filtered using a low-pass filter with a spatial frequency of 250 μm^{-1}, and $I(l, m)$ is the intensity of the corrected speckle field.

Despite in the current setup a conventional data processing approach is required to make the speckle fields spatially uniform, this can be completely compensated with a different choice of the setup. In particular one can use four microlenses close to the scatterer to overlap, on the observation plane, the centroids of the speckel fields generated from the four beams. In this way, the spatial normalization is not necessary. Alternatively one can replace the pyramid with a fiber optic splitter. In this case, the fibers can be oriented before the scattered in such a way the centroids of the speckle fields, generated from the four fibers, are overlapped on the observation plane after diffusion.

Note that the speckle fields acquired with the CCD (Fig. 4(b)) are characterized by the interference patterns, given by the superposition of the speckle fields from different inputs. The interference patterns have high contrast in a scale length typical of the transverse spatial coherence length $L_C \approx \lambda z/w$, where w is the transverse beam size on the scatterer, λ is the radiation wavelength and z is the distance between the CCD and the scatterer. The random behavior of the observed intensity that characterizes the speckle fields is of fundamental importance for the receptron implementation, being a consequence of the large amount of the representing parameters in Eq. (15), providing to the network high variability and non-linearity.

4.2. Non-linearity of the optical receptron

We checked the non-linearity of the optical receptron by applying the superposition principle to the uncorrected intensity of the speckle field. Since $I_{CCD}(\vec{x}) = S(\vec{x})$ (as discussed in Section 2.1) we calculated $I_D = I_{CCD}(1, 1, 1, 1) - [I_{CCD}(0, 0, 0, 1) + I_{CCD}(0, 0, 1, 0) + I_{CCD}(0, 1, 0, 0) + I_{CCD}(1, 0, 0, 0)]$. Non-linearity of receptron has been represented in Fig. 5 where we show I_D and the relative distribution. We observe that the Root Mean Square (RMS) of I_D (≈ 28, in a grey scale range between 255 and -255) is comparable with those of the other speckle patterns. Moreover, we exclude that the observed fluctuations originate from noise, since noise has substantially lower RMS ($\sigma_n \approx 0.42$), concluding that speckle patterns behave with the expected non-linearity with respect to the inputs. We exploit the non-linearity (square modulus of the field) shown in Eq. (12), resulting from the acquisition of the intensity (and not of the radiation field) with the CCD. More in general, this can be obtained with any detector sensible to the radiation intensity since the non-linearity is obtained from the intensity of the speckle field.

While the results shown in Fig. 5 prove that the proposed system truly implements the receptron model, we are interested in a more general approach to define the non-linear properties. In the case of the optical receptron the tensor is reduced to a matrix, so we can use some of the standard measures for diagonality of a matrix. In particular, we can measure the relative importance of off-diagonal versus diagonal elements via the Frobenius norm (Alyani, Congedo, & Moakher, 2017) for the optical receptron:

$$\mu = \frac{\|W - diag(W)\|_F}{\|W\|_F} = \left(\frac{2\sum_{j<k} w_{jk}^2}{\sum_j w_j^2 + 2\sum_{j<k} w_{jk}^2} \right)^{1/2} \tag{18}$$

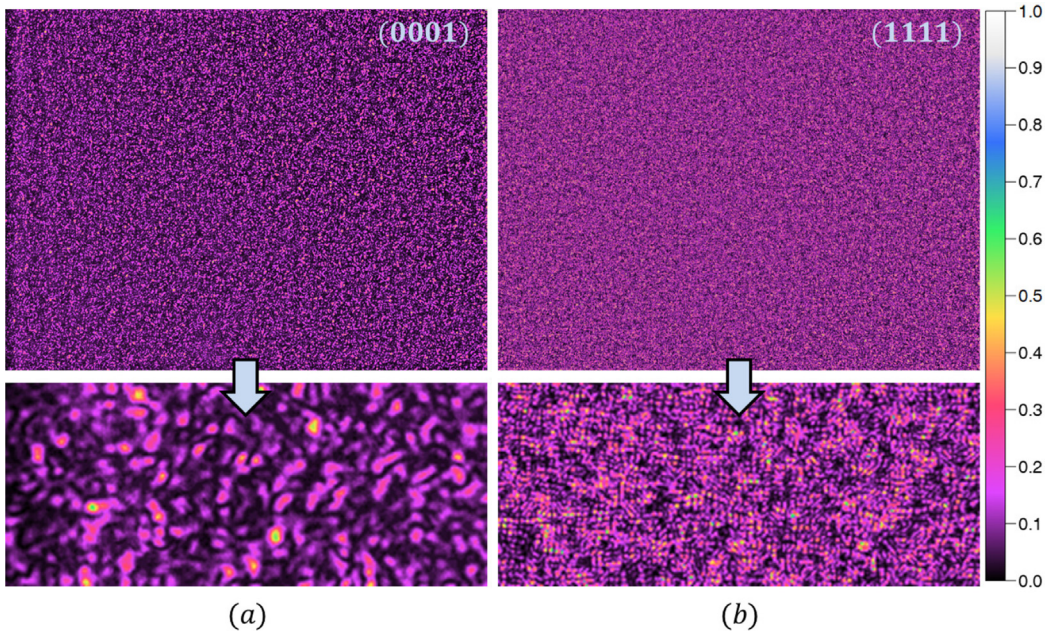

Fig. 4. Images of the speckle fields (in false colors) acquired with the CCD for the two input combinations $(x_1, x_2, x_3, x_4) = (0, 0, 0, 1)$ (a) and $(x_1, x_2, x_3, x_4) = (1, 1, 1, 1)$ (b). The higher density of the speckles in (b), is due to the formation of interference fringes created by the superposition of the speckle fields produced with the four input laser beams. The bottom panels are a zoom of a portion of the top images.

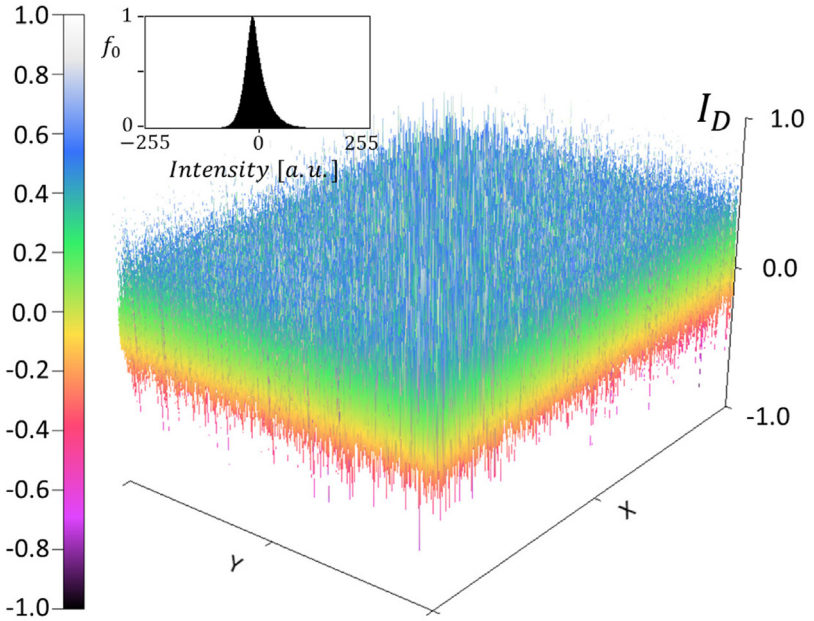

Fig. 5. 3D-view of the speckle field obtained with the difference I_D normalized to the maximum intensity. The inset shows the intensity distribution of I_D with the occurrence f_0 normalized to the maximum value. The RMS intensity ($I_{D(RMS)} = 28$) is not in agreement with the superposition principle for linear systems ($I_{D(RMS)} = 0$).

where W is the weights' tensor and $\|\cdot\|_F$ is the Frobenius norm. The matrix elements of the tensor have been extrapolated from data by inverting Eq. (5): the 2^n independent parameters for a given receptron are obtained from the system of 2^n equations by substituting into Eq. (5) all the possible input combinations e.g. $w_3 = I_{CCD}(0, 0, 1, 0) - I_{CCD}(0, 0, 0, 0)$, $w_{12} = I_{CCD}(1, 1, 0, 0) - I_{CCD}(1, 0, 0, 0) - I_{CCD}(0, 1, 0, 0) - I_{CCD}(0, 0, 0, 0)$. The coefficient μ lies between 0 and 1, where 0 indicates fully diagonal tensor and 1 a vanishing diagonal.

The high values achieved (see Fig. 6) demonstrate the remarkable capabilities of the optical system, which reproduces almost all levels of linearity of a theoretical receptron. Apart from confirming the result of Fig. 5, this example well demonstrates how the weights provide a powerful level of abstraction which is in turn an additional tool in the characterization of the system behavior.

4.3. Implementation and classification of the boolean functions

The implementation of different Boolean functions depending on the input combinations was verified experimentally with a set of 16 speckle fields as shown in Fig. 3(c). These fields correspond

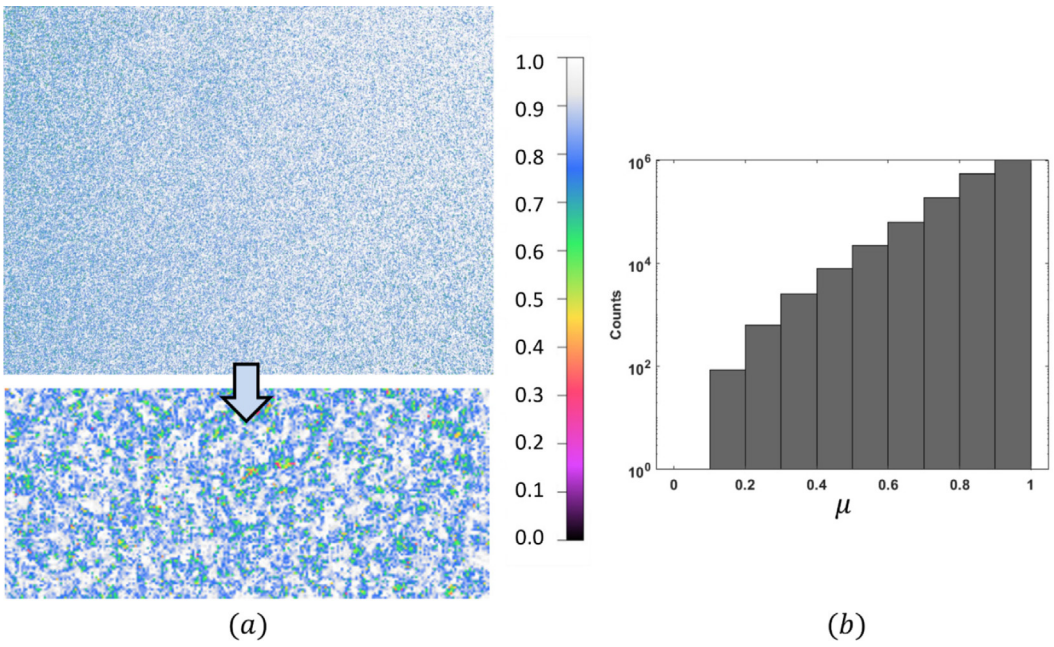

Fig. 6. (a) 2D-view of μ obtained for the same data shown in Fig. 5 (bottom panel is a zoom of the top left corner). (b) Distribution of μ in semilog scale. Notice that the μ value of the largest number of pixels are concentrated close to unity, showing the expected highly non-linear properties.

to the digital combination of inputs ranging from $\vec{x} = (0, 0, 0, 0)$ to $\vec{x} = (1, 1, 1, 1)$ (or between 0 and 15 in the decimal base system). The grey scale images are converted in black and white images (Boolean fields), by applying a threshold th in the range between 0 and 255 (8 bit grey scale range) in accordance with the thresholding process in Eq. (3). Note that each pixel of the CCD (in grey scale) represents an independent function S, thus the intensity $I(l, m) = S(l, m)$ reproduces a collection of $l \times m$ non-linear functions, characterized by high statistical variability. Therefore the thresholding process generates $l \times m$ digital outputs $Y(l, m)$, each implementing its own Boolean function.

Thanks to the high variability of the Boolean functions in the observation plane, the training of the network does not require an additional modification of the weights to solve a specific task. Network training is carried out for any function through the acquisition of 16 images, corresponding to the digital input combinations, and selecting the pixel (or pixels) that solve the target function regardless of the complexity of the function (see Fig. 3(c)). Therefore, the training procedure consists in associating the position of the Boolean field (pixel coordinates) with the target function.

An example of Boolean field obtained for $th = 55$ is shown in Fig. 7(a). Notice that the white pixels are grouped in correspondence with the speckle areas providing a strongly correlated statistic. The statistical properties of the Boolean fields generated, deeply linked with their variability, can be controlled acting on the speckle spatial coherence length L_C or on the threshold process. The result can be evaluated with the multiplicity, i.e. the number of high states, of the Boolean function generated, which corresponds to the number of success events of a binomial statistical process made of 2^n trials. The experimental distributions of the multiplicity of the output functions are shown in Fig. 8. Distributions are obtained on the sample of 1600×1200 pixels and for three different thresholds. Results can be compared with a binomial distribution, which is valid in the case of fully uncorrelated light intensities, having the success probability given by

$$p = \frac{\sum_k \int_{th}^{\infty} \rho_k(I) dI}{2^n - 1}, \tag{19}$$

where ρ_k is the normalized intensity distribution ($\int_0^{\infty} \rho_k(I) dI = 1$) of the speckle field with input combination k. The probability p for the binomial distribution is obtained considering that the different speckle fields, obtained for different input combinations k, have small differences in the intensity distributions and hence different probabilities p_k. We thus estimate the average of these probabilities by means of Eq. (19), where the integral $\int_{th}^{\infty} \rho_k(I) dI$ is the probability that the pixel intensity overcomes the threshold th with the speckle field produced by means of the input combination k. The average probability over the input combinations is therefore $\frac{\sum_k p_k}{2^n - 1}$, where n is the number of inputs. Notice that the number of possible combinations is $2^n - 1$ instead of 2^n since we omit the trivial combination (0, 0, 0, 0). From an experimental point of view $n = 4$ is the number of inputs (number of beams used as inputs) while k numbers the combinations ($1 \leq k \leq 15$): $k = 1$ implies $(x_1, x_2, x_3, x_4) = (0, 0, 0, 1)$, $k = 2$ implies $(x_1, x_2, x_3, x_4) = (0, 0, 1, 0)$, …, $k = 15$ implies $(x_1, x_2, x_3, x_4) = (1, 1, 1, 1)$.

The peak of the experimental distributions can be shifted at low, medium and high multiplicity for the three probabilities $p = 0,07$, $p = 0,48$, $p = 0,83$ acting on the threshold: this provides a method to promote the solution of functions with a given multiplicity. Note the deviations from the binomial model especially for the two middle scenarios: we interpret the difference as due to correlations generated by the coherence areas of the speckles, as also observed in the Boolean field of Fig. 7(a). The main assumption necessary for a good matching between the theory and the experiment is the independency of the statistical process both from the pixel position and from the input combination. We observed that the first condition is more critical, because the speckle field is always characterized by a certain degree of coherence that make the pixel strongly correlated in the coherence area of each speckle. The consequence is that white pixels and black pixels appear as clusterized in the Boolean fields. In other words, the statistical process is not independent on the pixel position since adjacent pixels are correlated.

To verify our interpretation, we generated a new set of 16 Boolean fields starting from the same experimental data by applying two thresholds $th_1 = 25$, $th_2 = 30$ (in a 8 bit grey scale

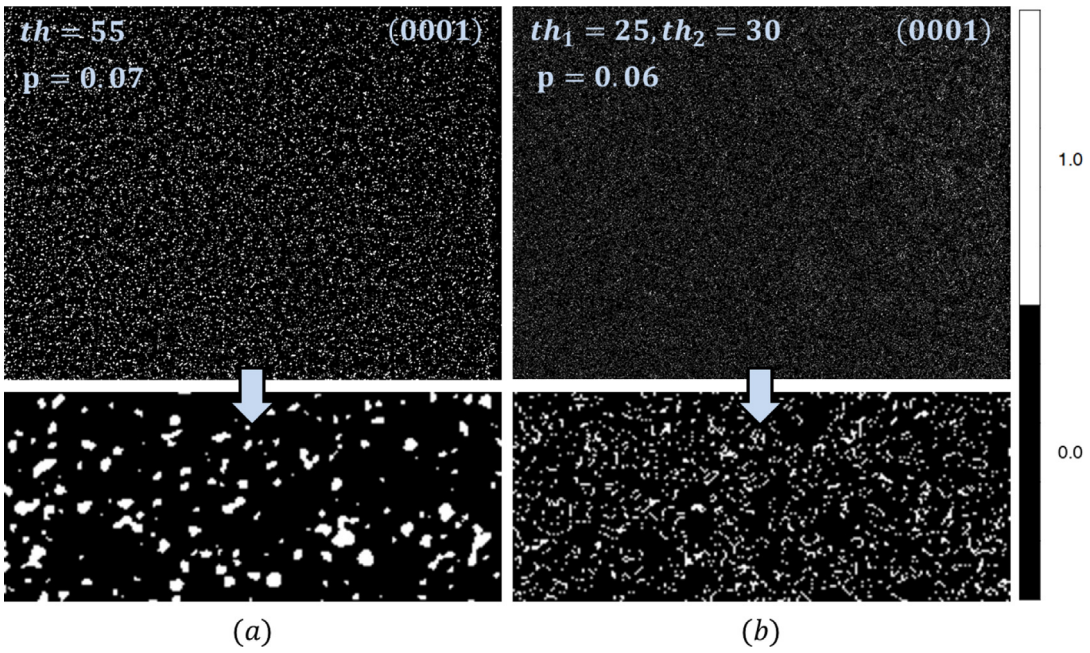

Fig. 7. Images of the Boolean fields generated with single threshold (a) and double threshold (b) for the input combination $(x_1, x_2, x_3, x_4) = (0, 0, 0, 1)$. Despite the average success probability p of (a) and (b) are similar, the double threshold drastically reduces the white speckles size. The bottom panels are a zoom of a portion of the top images.

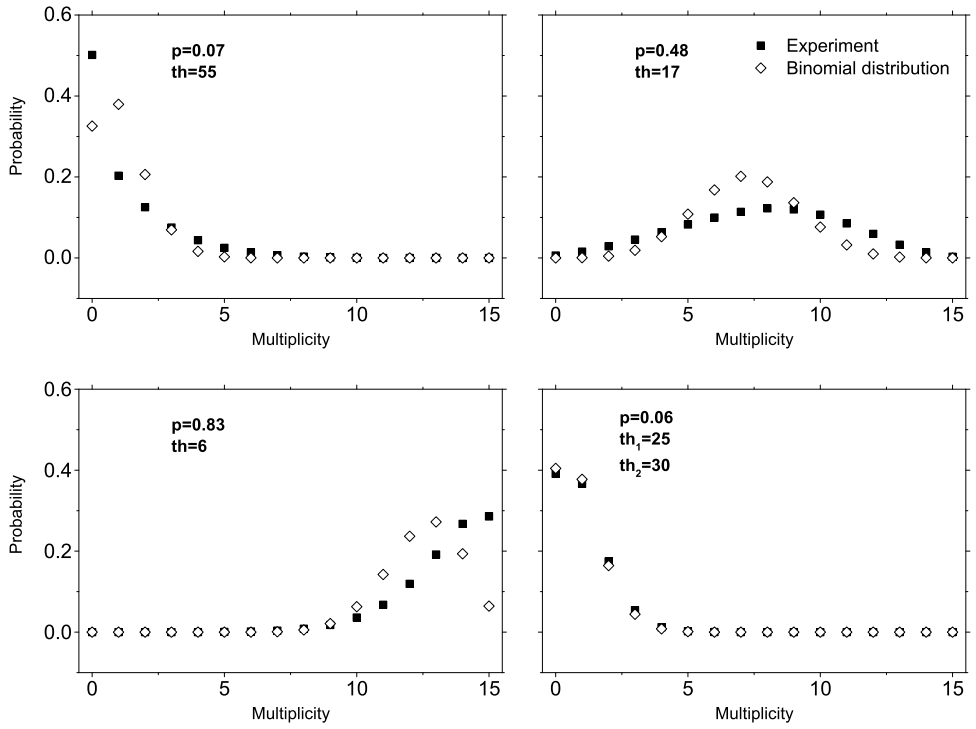

Fig. 8. Experimental distributions of the functions classified with the multiplicity for different probabilities p. Data are compared with the theoretical binomial distributions. The distribution obtained with the double threshold (right-bottom) is in good agreement with the theory thanks to the minimization of pixel correlations generated by the speckle fields.

range). The Boolean field was obtained by taking as "HIGH" state the pixel intensity of the corresponding speckle field in between the thresholds th_1, th_2 and by taking as "LOW" state the pixel intensity outside the two thresholds. Therefore, when the two thresholds are close each other, we expect a reduced speckles area composed of a few pixels, thus reducing the pixel correlation. By using a double threshold we experimentally observed that the Boolean field reduces the clustering (see Fig. 7(b)) and hence we assume that the pixel correlations are also reduced. Our assumption is confirmed by the good matching between the

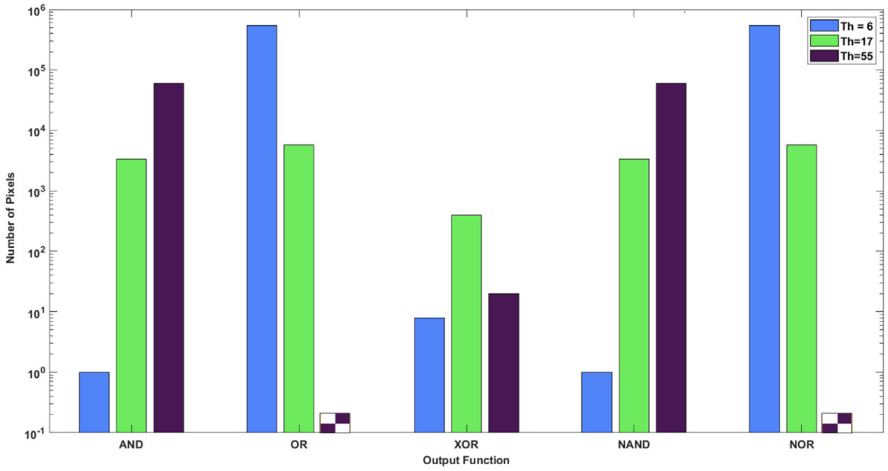

Fig. 9. Number of standard Boolean functions (AND, OR, XOR, NAND and NOR) solved by the network for different thresholds in logarithmic scale. Note that, due to complementary conditions shown in Eq. (21), the number of NOR and OR is the same, as well as for AND and NAND functions. The checkerboard bars for OR and NOR stand for zero pixels.

binomial and the experimental distributions, bottom right plot in Fig. 8, obtained with the double threshold. The experimental distribution is compared with the corresponding theoretical binomial distribution with success probability,

$$p = \frac{\sum_k \int_{th_1}^{th_2} \rho_k(I) dI}{2^n - 1}. \tag{20}$$

With the single threshold the clustering is always observed and hence we do not expect a matching between the theory and the experiment as also confirmed from the others distribution in Fig. 8. To reduce the clustering of the Boolean field we need two thresholds close together which implies a low probability p, not applicable to obtain the distributions of Fig. 8 with high probability. These considerations do not exclude the possibility to modulate the shape of the distributions around the mean value, for example by increasing (and not reducing) the clustering.

Summarizing, correlations can be used to change the shape of the multiplicity distribution, while the threshold gives control over the mean of such a distribution.

We have applied these tools to solve Boolean functions (Fig. 9): in particular, we show the number of pixels that implement the functions $Y(x_1, x_2, x_3, x_4) = x_1 \cdot x_2 \cdot x_3 \cdot x_4$ (four-input AND gate), $Y(x_1, x_2, x_3, x_4) = x_1 + x_2 + x_3 + x_4$ (four-input OR gate) and $Y(x_1, x_2, x_3, x_4) = x_2 \oplus x_3$ as a function of the chosen threshold. The three functions have multiplicities $m = 1$, $m = 15$ and $m = 8$, and are solved with higher efficiency (higher number of pixels) when the multiplicity is closer to the peak probability of the distributions shown in Fig. 8. It occurs for the three corresponding distributions having binomial success probabilities $p = 0.07$, $p = 0.83$, $p = 0.48$ and obtained with the thresholds $th = 55$, $th = 6$, $th = 17$, respectively. Similar considerations can be done to effectively solve the functions $Y(x_1, x_2, x_3, x_4) = \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4}$ (NAND universal gate) and $Y(x_1, x_2, x_3, x_4) = \overline{x_1 + x_2 + x_3 + x_4}$ (NOR universal gate). In fact, as shown in Fig. 9 we obtain the same number of pixels of the corresponding negated functions AND, OR by using the complementary conditions of Eq. (3) as

$$Y(x_1, \ldots, x_n) = \begin{cases} 1 & S \leq th \\ 0 & S > th\,. \end{cases} \tag{21}$$

These results confirm that by controlling the statistical properties of the resulting intensities we drastically increase the computing efficiency for a given Boolean algebraic problem. Moreover, the non-linear behavior experimentally verified in Section 4.2 makes it possible to solve non-linearly separable functions, as formally described in the example of Fig. 1. The experimental result in Fig. 9 relative to XOR operator gives a direct proof of the receptron advantages with respect to the perceptron in terms of the type of generated functions.

5. Conclusions

We have developed a formal description of a device, the receptron, which generalizes the perceptron by considering that the input weights are not univocally related to a single input, hence they cannot be independently adjusted. This non-linear characteristic is more similar to what observed in the interactions between synapses in neural dendritic trees (Bicknell & Häusser, 2021).

An optical implementation of a receptron has been experimentally realized: the setup exploits simple optical elements to generate a large number of Boolean functions ($\approx 2 \cdot 10^6$) by using the high variability of the speckle fields. Although a benckmark between different technologies appears difficult, we provide a comparison with standard CMOS technology in terms of effective number of implemented 4-input gates, that in our case is of the order of 10^6. A similar number can be realized with standard Very Large Scale Integration (VLSI) technology. The number of effective gates is of course not sufficient to assess the computing efficiency, for instance the computing speed plays a fundamental role in the overall efficiency of a computing device. Our approach is fully compatible with existing optical technologies, typically used in optical telecommunications, which achieve clock frequencies higher than tens of GHz. However, further developments are required to assess the overall efficiency by simultaneously taking into account the integration scale and the computing speed.

The complete set of functions has been classified with the multiplicity, in order to characterize their statistical properties. The role of the macroscopic parameters, such as the activation threshold, is essential to statistically control these properties. In fact, by changing the threshold the average multiplicity of the distributions can be changed in a controlled way. Furthermore, a control of the distribution shapes can be also managed by properly setting the double threshold in the thresholding process. These results show a method for solving (by shifting the peak probability or by changing the shape of the distribution) particular classes of Boolean functions and hence to statistically promote the solution of particular Boolean algebra problems.

The support of a conventional computer was necessary in this proof-of-principle experiment to perform the thresholding

process and the initial training procedure. Once these steps are realized, the optical system becomes independent on any conventional computer. For example, the speckle fields in correspondence of the CCD pixels, of one or more target functions, can be directly coupled with one or more photodiodes and amplifiers. In this way, the thresholding procedure can be realized with simple comparator circuits, by properly matching the computational and comparator thresholds. Note that the CCD and the photodiodes can coexist in the same optical setup, since the observation plane can be easily duplicated with simple optical elements (e.g. beam splitters or mirrors). This method drastically increases the computing speed, being substantially limited only by the time responses of photodiodes, amplifiers and comparators with working frequencies above tens of GHz.

The search of a defined function with a suitable conventional algorithm is performed by the use of a conventional computer or a microcontroller. In this case, the efficiency in term of computing speed depends of the device used. At the end of the process we know that a given pixel (or group of pixels) will provide the target function when the threshold is applied to its generated intensity. The threshold process is an operation that can be realized with simple comparators.

The optical implementation of the receptron scheme opens the way for the fabrication of a completely new class of optical devices for neuromorphic data processing based on a very simple hardware: a single receptron is already capable of solving nonlinearly functions, therefore a network of receptrons could allow the combination of the outputs of several units to obtain the target function easier and faster. In stark contrast with a perceptron network, an optical receptron network would not intrinsically require more energy than its single-element counterpart. In other words, a receptron can already generate all functions in principle, but the search for a desired function could take too long, due to the exponential increase in the number of possibilities as a function of the inputs. A simple combination of receptrons could instead simplify the search. Since the number of parameters is very large compared to a perceptron, one does not need a fine control of each parameter to obtain a given result, since it can be obtained with different configurations of the same parameters. These aspects are of fundamental interest in view of the fabrication and scale-up of optical receptron networks for very complex data processing tasks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Alyani, K., Congedo, M., & Moakher, M. (2017). Diagonality measures of hermitian positive-definite matrices with application to the approximate joint diagonalization problem. *Linear Algebra and its Applications*, *528*, 290–320.

Ambrogio, S., Narayanan, P., Tsai, H., et al. (2018). Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature*, *558*, 60–67.

Bicknell, B. A., & Häusser, M. (2021). A synaptic learning rule for exploiting nonlinear dendritic computation. *Neuron*, *109*, 4001–4017.

Burr, G. W., Shelby, R. M., Sebastian, A., Kim, S., Kim, S., Sidler, S., et al. (2016). Neuromorphic computing using non-volatile memory. *Advances in Physics: X*, *2*, 89–124.

Dainty, J. C. (1975). *Laser speckle and related phenomena*. Heidelberg: Springer-Verlag Berlin.

Diaz-Alvarez, A., Higuchi, R., Sanz-Leon, P., et al. (2019). Emergent dynamics of neuromorphic nanowire networks. *Scientific Reports*, *9*, 14920.

Goodman, J. W. (2007). *Speckle phenomena in optics*. Englewood: Ben Roberts & Company.

Häusser, M., Spruston, N., & Stuart, G. J. (2000). Diversity and dynamics of dendritic signaling. *Science*, *290*, 739–744.

Hochstetter, J., Zhu, R., Loeffler, A., et al. (2021). Avalanches and edge-of-chaos learning in neuromorphic nanowire networks. *Nature Communication*, *12*, 4008.

Jeong, D. S., Kim, I., Ziegler, M., & Kohlstedt, H. (2013). Towards artificial neurons and synapses: a materials point of view. *RSC Advances*, *3*, 3169–3183.

Li, Q., Diaz-Alvarez, A., Iguchi, R., Hochstetter, J., Loeffler, A., Zhu, R., et al. (2020). Dynamic electrical pathway tuning in neuromorphic nanowire networks. *Advanced Functional Materials*, *30*, Article 2003679.

Lillicrap, T. P., Santoro, A., Marris, L., et al. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, *21*, 335–346.

Liu, D., Yu, H., & Chai, Y. (2021). Low-power computing with neuromorphic engineering. *Advanced Intelligent System*, *3*, Article 2000150.

Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, *3*, 127–149.

Lynn, C. W., & Bassett, D. S. (2019). The physics of brain network structure, function and control. *Nature Reviews Physics*, *1*, 318–332.

Mallinson, J. B., Shirai, S., Acharya, S. K., Bose, S. K., Galli, E., & Brown, S. A. (2019). Avalanches and criticality in self-organized nanoscale networks. *Science Advances*, *5*, 1.

Martini, G., Mirigliano, M., Paroli, B., & Milani, P. (2022). The receptron: a device for the implementation of information processing systems based on complex nanostructured systems. *Japanese Journal of Applied Physics*, *61*, SM0801.

McCulloch, W. S., & Pitt, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133.

Milano, G., Miranda, E., & Ricciardi, C. (2022). Connectome of memristive nanowire networks through graph theory. *Neural Networks*, *150*, 137–148.

Milano, G., Pedretti, G., Fretto, M., Boarino, L., Benfenati, F., Ielmini, D., et al. (2020). Brain-inspired structural plasticity through reweighting and rewiring in multi-terminal self-organizing memristive nanowire networks. *Advanced Intelligent System*, *2*, Article 2000096.

Minsky, M., & Papert, S. (1970). A review of perceptrons: An introduction to computational geometry. *Information and Control*, *17*, 501–522.

Mirigliano, M., Decastri, D., Pullia, A., Dellasega, D., Casu, A., Falqui, A., et al. (2020). Complex electrical spiking activity in resistive switching nanostructured au two-terminal devices. *Nanotechnology*, *31*, Article 234001.

Mirigliano, M., & Milani, P. (2021). Electrical conduction in nanogranular cluster-assembled metallic films. *Advances in Physics: X*, *6*, Article 1908847.

Mirigliano, M., Paroli, B., Martini, G., Fedrizzi, M., Falqui, A., Casu, A., et al. (2021). A binary classifier based on a reconfigurable dense network of metallic nanojunctions. *Neuromorphic Comput. Eng.*, *1*, Article 024007.

Mirigliano, M., Radice, S., Falqui, A., et al. (2020). Anomalous electrical conduction and negative temperature coefficient of resistance in nanostructured gold resistive switching films. *Scientific Reports*, *10*, 19613.

Nagy, G. (1991). Neural networks - then and now. *IEEE Transactions on Neural Networks*, *2*, 316–318.

Nakajima, M., Tanaka, K., & Hashimoto, T. (2021). Scalable reservoir computing on coherent linear photonic processor. *Communications Physics*, *4*, 20.

Pershin, Y. V., & Di Ventra, M. (2010). Experimental demonstration of associative memory with memristive neural networks. *Neural Networks*, *23*, 881–886.

Poirazi, P., & Mel, B. W. (2001). Impact of active dendrites and structural plasticity on the memory capacity of neural tissue. *Neuron*, *29*, 779–796.

Rafayelyan, M., Dong, J., Tan, Y., Krzakala, F., & Gigan, S. (2020). Large-scale optical reservoir computing for spatiotemporal chaotic systems prediction. *Physical Review X*, *10*, Article 041037.

Rajendran, B., Sebastian, A., Schmuker, M., Srinivasa, N., & Eleftheriou, E. (2019). Low-power neuromorphic hardware for signal processing applications: A review of architectural and system-level design approaches. In *IEEE signal processing magazine, Vol. 36* (pp. 97–110).

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408.

Schuman, C. D., Kulkarni, S. R., Parsa, M., Mitchell, J. P., Date, P., & Kay, B. (2022). Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, *2*, 10–19.

Shen, Y., et al. (2017). Deep learning with coherent nanophotonic circuits. *Nature Photonics*, *11*, 441–446.

Silver, R. (2010). Neuronal arithmetics. *Nature Reviews Neuroscience*, *11*, 474–489.

Ting-Ho Lo, J. (2011). A low-order model of biological neural networks. *Neural Computation*, *23*, 2626–2682.

Tomson, D., et al. (2016). Roadmap on silicon photonics. *Journal of Optics*, *18*, Article 073003.

Tononi, G., Boly, M., Massimini, M., & Koch, C. (2016). Integrated information theory: from consciousness to its physical substrate. *Nature Reviews Neuroscience*, *17*, 450–461.

Vandoorne, K., Mechet, P., Van Vaerenbergh, T., et al. (2014). Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature Communication*, *5*, 3541.

Xia, Q., & Yang, J. J. (2019). Memristive crossbar arrays for brain-inspired computing. *Nature Materials*, *18*, 309–323.