

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/340257021>

# Neurons as Pattern Recognizers (CDT-25)

Preprint · March 2020

DOI: 10.13140/RG.2.2.35407.43686/2

CITATIONS

0

READS

1,047

1 author:



[Luciano da F. Costa](#)

University of São Paulo

734 PUBLICATIONS 13,325 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Análise, Simulação e Estimação de Equações Diferenciais Parciais para o Desenvolvimento Embrionário [View project](#)



Biomedical and bioengineering [View project](#)

# Neurons as Pattern Recognizers (CDT-25)

Luciano da Fontoura Costa  
*luciano@ifsc.usp.br*

*São Carlos Institute of Physics – DFCM/USP*

v 1.1 : 17th April 2020  
v 1.0 : 25th March 2020

## Abstract

In this work we address the interpretation of neuronal activity as performing pattern recognition. After briefly describing biological neuronal dynamics, we present a simplified model of a neuronal cell involving linear combination of inputs followed by a hard-limiting non-linearity. Then, two possible interpretations of neurons are discussed: as linear discriminators, and as template matchers. The concept of linear pattern morphing is described as a preparation for addressing a type of artificial neuronal network for non-supervised learning known as self-organized maps (SOMs), which is then presented and illustrated.

“Mientras el cerebro sea un misterio, el universo continuará siendo un misterio.”

---

*Santiago Ramón y Cajal.*

## 1 Introduction

Unicellular organisms... multicellular organisms... neuronal networks... science!

As with cosmic matter, cells – the atoms of biology – presents a tendency to come together, giving rise to organisms capable of more flexible and intricate behavior. The neuronal networks underlying the nervous system represent a further step along this trend. By possessing prolongations (dendrites and axons), neuronal cells became specialized in making connections, necessary for transmitting and processing information. The thus obtained information processing systems, capable of impressive performance, ultimately gave rise to human society, culture and science. Even though not often realized, much of these activities actually corresponds to instances of *pattern recognition*, informally the task of assigning new or previously defined categories to world entities. The very reading of these lines provides a related example, as we associate meaning to each word and sentence.

The reference performance of biological neuronal networks have continuously motivated efforts from the scientific community in developing *artificial neuronal networks*

having as main objectives: (i) to model (e.g. [1]) and better understand biological neuronal networks; and (ii) to develop systems that can be applied in several pattern recognition and artificial intelligence tasks.

The present work is aimed at presenting and discussing neuronal cells in one of their most important roles: pattern recognition. Here, after briefly presenting the biological operation of neurons, we describe a simple respective mathematical model that allows us to address two possible respective interpretations of neuronal action as pattern recognition: neurons as linear discriminators, and neurons as template matchers.

Next, we introduce the concept of pattern linear morphing, corresponding to interpolations between two given patterns. These principles are then adopted to describe and illustrate an interesting artificial neuronal network model, namely *self-organizing maps* – SOMs. In these networks, which are primarily intended for non-supervised pattern recognition, each linear neuron in an artificial ‘*cortex*’ acts as a pattern recognizer, receiving information from the whole pattern being presented, changing not only its weights, but also the weights of other nearby neurons, so as to become more similar to the present input. This dynamics, which incorporates interesting biological principles such as topographic cortical organization and function similarity between nearby cells, can result in self-organized maps, with regions associated to the main types of presented patterns. Interestingly, trajectories along the

‘cortex’ can often be understood as morphings between the identified categories of patterns.

By corresponding to a type of convolutional artificial neuronal networks, SOMs are closely related to *deep learning* (e.g. [2]), therefore providing an interesting entry point to this impressive area.

## 2 A Brief Review of Neuronal Dynamics

Neurons can be thought of as biological cells specialized for information transmission and processing. Indeed, while many biological cells have nearly round shapes, neurons are characterized by prolongations (axons and dendrites) allowing them to interconnect as networks. Being themselves cells, much can be known about neurons by studying respective biological principles (e.g. [3]).

A cell is a biological entity delimited by a respective *membrane* (bilayer) composed of phospholipids, which has high electrical impedance and blocks most matter flow except for some small ions. Cells can exchange information, mass and energy through proteins that goes through the cell membrane and act as *transmembrane channels*. These channels can open and close under control of electric potential, other molecules, etc. Some transmembrane proteins act as *ion pumps*, transferring ions in or out the cell interior. Such pumping requires energy consumption.

As a consequence of charge and osmotic balance through the membrane, a negative electric rest (or equilibrium) potential of approximately  $-60mV$  is established inside the cell with respect to its exterior. Any increase of this voltage is typically understood as a *depolarization*.

As cells, neurons incorporate all the above structures and mechanisms. In addition to that, neurons have two main types of tree-like extensions: (i) *dendrites*, which receive information; and (ii) *axon*, which transmit information to other cells. The diagram in Figure 1 illustrates a highly simplified neuronal cell, including cell soma (or perikarion), dendrites, axon, and synapses.

The connection between two neuronal cells is accomplished through *synapses*, incorporating the pre-synaptic button (at the axons extremities), the synaptic cleft (the synaptic interstice between the two cells), and the post-synaptic cell membrane. The action of a synapsis is mainly controlled by the action potential it can receive from the respective axon. The *action potential* is characterized by *spikes* or *impulses* of electric depolarization.

Once a synapsis start receiving spikes, it releases *neurotransmitters* into the cleft, which act on the postsynaptic transmembrane channels, allowing intake of ions such as  $Na^+$ . Being positive, these ions depolarize the postsy-

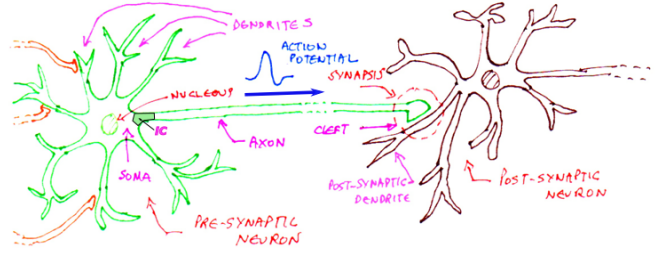


Figure 1: The basic components of a neuronal cell include its cell body (*soma*) and respective *nucleous*, *dendrites*, *axon* and *synapses*. A synaptic connection established by a given cell define a reference for the *pre-synaptic* neuron (the same cell) and the *post-synaptic* neuron (the cell that is being contacted). Input signals are received at the dendrites, and the neuronal output corresponds to electric pulses, known as *action potentials*, that propagate along the axon until the respective synaptic terminations. These action potentials originate at the implantation cone (IC), once its undergoes a depolarization of about  $-60mV$ .

naptic region close to the respective synapsis, and this depolarization then propagates through diffusion along the dendrites and soma. The depolarizations emanating from several synapses are combined, and when the electric potential at the post-synaptic *implantation cone* (IC) increases by approximately  $20mV$  (typically), voltage-controlled transmembrane channels open, allowing intake of  $Na^+$ , which further depolarizes the implantation cone and beginning of the axon. This implies an abrupt change of the local cell potential, giving rise to an action potential.

Depolarization can be transmitted through the axon internal as well as externally, inducing a chain reaction along the axon, corresponding to an *action potential*. As the action potential ceases locally, ion pumps on the implantation cone and axon act in order to release the excess  $Na^+$ , which is done at the expense of energy consumption. The presence of *myelin sheaths* along the axons of several types of neurons can act as an insulator, promoting saltatory conduction through voltage fields external to the axon, minimizing energy consumption, as only the pumps between sheaths (nodes of Janvier) need to operate, also implying faster action potential transmission.

The effect of a pre-synaptic action potential on the subsequent post-synaptic neuron depends on the synaptic efficiency, in the sense that the higher this efficiency the higher the respectively implied depolarization. Synaptic efficiency is largely influenced by the previous history of joint activation of the two involved cells, mediated by *neuromodulators*. Observe that a great extent of biological *memory* corresponds to the respective synaptic weights, managed by neuromodulators.

This is only an approximate and generic description of the neuronal dynamics, which involves many other details

and can vary in several manners and cases.

### 3 A Simple Neuronal Model

The above described principle of neuronal action can be modeled at several levels of complexity. Figure 2 depicts a relatively simple, mesoscale mathematical model of a neuron.

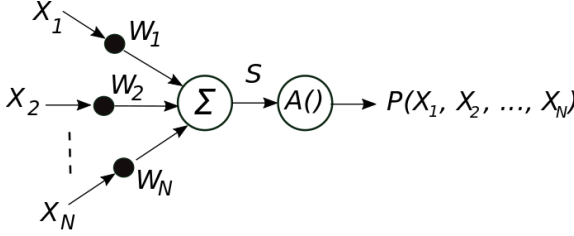


Figure 2: A simple model of the biological action of a neuron as revised in Section 2. The linear combination of the depolarizations implied by active synapses into the implantation cone is modeled in terms of the inner product  $\langle \vec{X}, \vec{W} \rangle = \sum_{i=1}^N X_i W_i$ , which is then fed into a nonlinear hard-limiting function  $A()$ , which models the abrupt generation of action potentials in the implantation cone, yielding the respective output  $P(X_1, X_2, \dots, X_N)$ .

The dendrites are represented by input connections associated to each of the input values  $X_i$ ,  $i = 1, 2, \dots, N$ , which can be understood as the average action potential excitation received from incoming axons. These intensities are multiplied by respective weights  $W_i$ , which correspond to the respective synaptic efficiencies. The depolarizing effect of these input is linearly integrated, implementing the respective inner product  $S = \langle \vec{X}, \vec{W} \rangle = \sum_{i=1}^N X_i W_i$ , modeling the the depolarization of the implantation cone as a consequence of the combined effect of diffusion of the depolarizations at the synapses.

The integrated activation  $S$  is then fed to a hard-limiting function  $A()$ , modeling the threshold operation at the implantation cone. A sigmoid function such as that in Equation 1 can be used for this finality.

$$P = A(S) = \frac{1}{1 + e^{-\beta S}} \quad (1)$$

where  $\beta$  is a parameter that controls the smoothness of the sigmoid transition, becoming sharper for larger values of  $\beta$ . Observe that there are other types of sigmoids and activation functions that can be adopted. It is possible to consider a simplification of this model in which the non-linear part is omitted, yielding a respective a *linear neuron*.

### 4 The Neuron as a Linear Discriminator

One possible interpretation of neuronal action is as *linear discrimination*. Consider the situation shown in Figure 3(a).

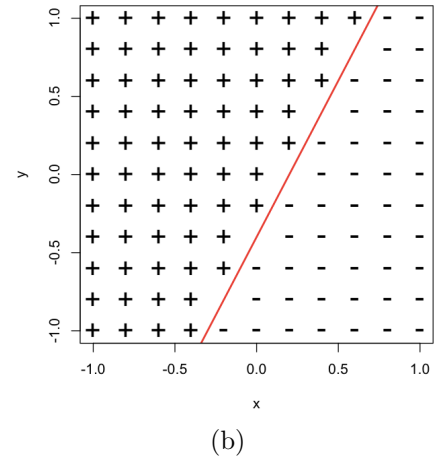
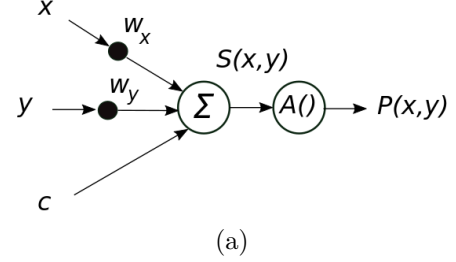


Figure 3: The action of a two input neuron as a linear discriminator: any point  $(x, y)$  to the lefthand side of the *separatrix* (red line) will yield positive sign, while those to the righthand side will be negative. This example considers  $c = 0.2$ ,  $w_x = -1$ , and  $w_y = 0.5$ , but other configurations of  $w_x$  and  $w_y$  would yield the same result as in this figure, for there is an infinite number of plans associated to the same discriminant line in red.

This single neuron has three inputs:  $x$ ,  $y$  and a constant signal  $c$ . Therefore, it will produce as result

$$P(x, y) = A(xw_x + yw_y + c) \quad (2)$$

It is interesting to find the set of points  $(x, y)$  that makes  $xw_x + yw_y + c$  equal to zero:

$$0 = xw_x + yw_y + c \rightarrow y = -\frac{c + xw_x}{w_y} \quad (3)$$

Observe that the above result corresponds to a line equation, defining a respective line in the  $(x, y)$  space as illustrated in red in Figure 3.

Figure 3(b) depicts the input space  $(x, y)$  as well as the sign of  $P(x, y)$  obtained in this space. It can be verified that the neuron can identify, through the sign of  $P(x, y)$ , in which of the two half-planes any instance  $(\tilde{x}, \tilde{y})$  of the

signal is. For instance, any input above the separating line  $L$  will yield  $+$  sign, and any input below that line will imply  $-$  sign.

In the case of a neuron as above receiving higher dimensional input, e.g.  $\mathbb{R}^N$ , the neuron will also implement linear discrimination, but respectively to a *hyperplan* determined by its respective parameters (weights and bias). These neurons can also be organized into successive layers, as in the artificial neural network model known as *perceptron* (e.g. [4]), then becoming capable of more general decision hypersurfaces.

The above characterized action of neurons as linear discrimination can immediately be applied to pattern recognition. For instance, in the case in Figure 3, each of the  $+$  and  $-$  regions can be associated to two different categories, so that a new point mapped at  $(x, y)$  will be assigned the respective sign as its recognized category.

## 5 The Neuron as a Template Matcher

Let  $\vec{u} = [u_x, u_y]^T$  and  $\vec{v} = [v_x, v_y]^T$  be two vectors in  $\mathbb{R}^2$ . Their inner (or scalar) product can be defined as the following scalar value:

$$\langle \vec{u}, \vec{v} \rangle = u_x v_x + u_y v_y = |\vec{u}| |\vec{v}| \cos(\theta) \quad (4)$$

where  $\theta$  is the smallest angle between these two vectors.

Figure 4 illustrates the two vectors  $\vec{u}$  and  $\vec{v}$ .

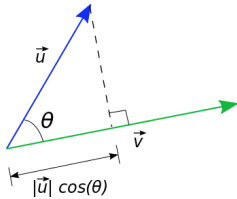


Figure 4: Two vectors  $\vec{u}$  and  $\vec{v}$  and the smallest angle  $\theta$  between them. The respective inner product  $\langle \vec{u}, \vec{v} \rangle$  can be understood as the magnitude of the projection of one vector into the other (e.g.  $|\vec{u}| \cos(\theta)$ ) multiplied by the latter magnitude (e.g.  $|\vec{v}|$ ).

The magnitude of the projection of  $u$  into  $v$  can be immediately obtained by applying trigonometry as  $|\vec{u}| \cos(\theta)$ , implying that the vector product between those two vectors corresponds to the value obtained by multiplying the magnitude of the projection of one vector by the magnitude of the other vector.

Therefore, if the magnitudes  $|\vec{u}|$  and  $|\vec{v}|$  are both kept constant, the inner product will achieve its maximum value when the projection of  $\vec{u}$  into  $\vec{v}$  (or, alternatively,  $\vec{v}$  into  $\vec{u}$ ) will be maximum, which is achieved when both vectors are parallel and  $\theta = 0$ .

When the magnitude of the projection of one vector into the other becomes null ( $\theta = \pi/2$ ), the respective inner product equals zero and the vectors are said to be *orthogonal*. However, this is not the minimum inner product value, which is rather achieved when the two vectors are colinear but in opposite directions ( $\theta = \pi$ ).

A particularly interesting interpretation of the inner product between two vectors is as a measurement of the *similarity* between those two vectors. The similarity will be highest when the projection is maximum ( $\theta = 0$ ), and lowest when the vectors are opposed ( $\theta = \pi$ ).

The above interpretation hold also when the vectors are in higher dimensional spaces. In particular, the understanding of the inner product of two vectors as a measurement of similarity provides one of the keys for better understanding neuronal networks in a more conceptual manner. This can be immediately achieved by taking the synaptic input signal as one of the vectors (e.g.  $\vec{u}$ ) and the set of synaptic weights as another vector (e.g.  $\vec{v}$ ).

Figure 5 illustrates the action of a simple neuron as a template matcher.

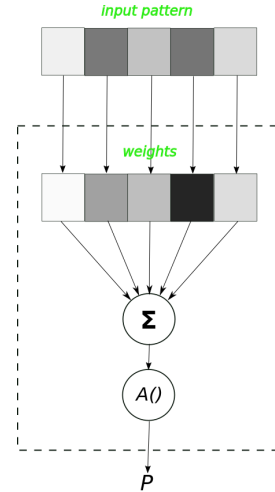


Figure 5: A simple neuron, with 5 inputs, acting as a template matcher by taking the inner product between the input values and respective weights, shown in gray levels. A relatively high output  $P$  is expected in case the signal resembles the signal, as in the present case.

Let's first consider a linear neuron, i.e. without the hard limiting function. The respective output corresponds precisely to the inner product between the input signal and the synaptic weights. Thus, the neuron will exhibit strong output whenever the input signal resembles its respectively associated weights. When the hard limiting non-linearity is included, the neuron will be active only when this similarity exceeds the respective threshold.

In a sense, we can say that any of these two types of neurons perform template matching (a kind of pattern recognition), yielding high output whenever the signal is

recognized as correspondingly closely to its weights. The conceptual potential of this interpretation will be illustrated when we discuss self-organizing maps in Section 7.

However, this approach has an intrinsic shortcoming in the sense that the inner product depends strongly on both the magnitudes of the two involved vectors. In case  $\vec{u}$  represents the synaptic weights, and therefore has fixed magnitude, a signal very distinct from the weights, but with large magnitude, can still induce a strong (false) indication of matching.

The above discussed template matching operation of a neuron is the basis for *convolutional* neuronal networks (e.g. [2]). Indeed, this follows from the fact that the convolution between two signals (e.g. [5]) can be understood as performing inner products between one of the signals and a domain-reversed version of the other signal.

In the next section we will discuss this problem in more detail and also present a possible simple and effective alternative scheme.

## 6 Interpolating Between Patterns

Given two patterns  $\vec{u}$  and  $\vec{v}$ , it is possible to interpolate patterns between them, an operation that is often called *morphing* of one pattern into the other. Interestingly, linear morphing is not only interesting and with application potentials, but it is also simple to be implemented by using the following expression:

$$\vec{r}(\alpha) = \vec{u} + \alpha(\vec{v} - \vec{u}) \quad (5)$$

allowing vector  $\vec{u}$  to be progressively morphed, according to increasing values of  $\alpha \in [0, 1]$ , into the other given vector  $\vec{v}$ . Figure 6 illustrates this basic principle of linear morphing from a vector  $\vec{u}$  into another vector  $\vec{v}$ .

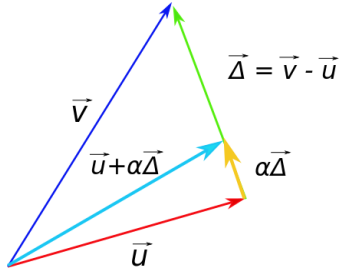


Figure 6: Given two vectors  $\vec{u}$  and  $\vec{v}$ , it is possible to obtain linear interpolations from  $\vec{u}$  to  $\vec{v}$  by obtaining the vector difference  $\vec{v} - \vec{u}$  and defining the interpolated vectors as  $\vec{r}(\alpha) = \vec{u} + \alpha\vec{v}$  for some  $\alpha \in [0, 1]$ .

An example of pattern morphing is presented in Figure 7, where the first pattern (top, left) is progressively morphed into the last pattern (bottom, right).

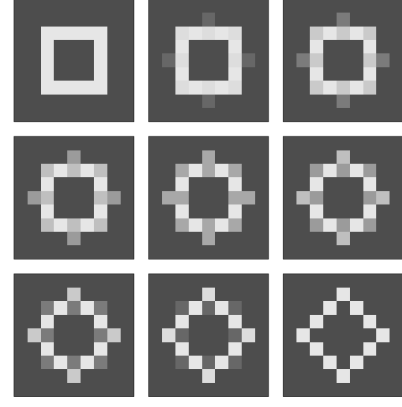


Figure 7: Stages along the linear morphing of the first pattern (top, left) into the last pattern (down, right) by using Eq. 5.

## 7 Self-Organizing Maps – SOMs

Self-organized maps, SOMs for short, are artificial neuronal networks (e.g. [4]) developed from the 70's, especially by T. Kohonen (e.g. [6]), as artificial neuronal networks for non-supervised classification (clustering). These networks are aimed at an increased neurobiological realism in the sense that the neurons are spatially organized along a typically 2D cortex and that each neuron influences other neurons in its neighborhood so that their properties become more similar, a property often observed among neurons in biological cortices (e.g. [7]).

Figure 8 shows the overall structure of one a simple SOM network.

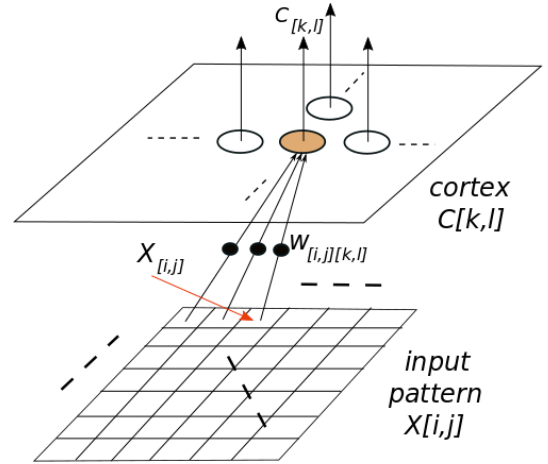


Figure 8: The overall organization of a SOM network. An  $N \times M$  input pattern is presented to an  $Q \times R$  neuronal network ('cortex'). Each linear neuron  $[k, l]$  receives input from all components  $[i, j]$  of the input pattern, through respective weights  $W_{[i,j][k,l]}$ , yielding respective output  $C_{[k,l]}$  corresponding to the inner product between the presented pattern and the respective weight, therefore providing an indication of the similarity between these two vectors.

The network itself consists of a 'cortex' containing  $Q \times R$

linear neurons, which are typically organized as a lattice  $[k, l]$ ,  $i = 1, 2, \dots, Q$ ;  $j = 1, 2, \dots, R$ . Interestingly, there is no interconnection between the neurons in the ‘cortex’, as their interaction is implemented during the learning stage. In this configuration, the patterns correspond to  $N \times M$  matrices (other types of patterns are also possible), each pixel  $X_{[i,j]}$  being fed into all neurons in the ‘cortex’, scaled by respective weight  $W_{[i,j][k,l]}$ . Therefore, the output of each neuron can be thought as corresponding to the inner product between the pattern and the respective weight, yielding an indication of the similarity between these vectors (observe that matrices and images can be transformed into respective vectors by row or column stacking).

Though aimed at non-supervised pattern recognition, typically a SOM will require *training* and subsequent *recognizing* stages. In the training stage, patterns are presented to the network, which updates its weights according to the learning rule illustrated in Figure 9.

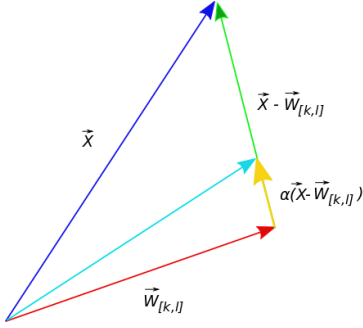


Figure 9: The learning rule in a typical SOM. Given a current pattern  $\vec{X}$  being currently presented, a neuron  $[k, l]$  can be biased towards recognizing it by making its respective weights  $\vec{W}_{[k,l]}$  more similar to the pattern. Mathematically, this operation is analogous to the morphing of the weight vector into the input pattern (see Fig. 6), yielding a modified weight  $\vec{W}_{[k,l]} = \vec{W}_{[k,l]} + \alpha(\vec{X} - \vec{W}_{[k,l]})$ , where  $\alpha$  is a learning rate belonging to the interval  $[0, 1]$ . The higher the value of  $\alpha$ , the more similar the new weight will be to the input pattern  $\vec{X}$ .

Basically, for each neuron  $[k, l]$ , the vector difference between the input pattern and the respective weight  $[\vec{W}_{[k,l]}]$ , yielding  $(\vec{X} - \vec{W}_{[k,l]})$ , is obtained and then multiplied by the learning rate  $\alpha$  and added to  $[\vec{W}_{[k,l]}]$ , with  $\alpha \in [0, 1]$ . Observe that this operation makes the weight of the considered neuron more similar to the input pattern. Indeed, in case  $\alpha$  were 1, the weight would become identical to the input pattern. Smaller values of  $\alpha$  are often used in order to cater for robustness and generalization during the pattern recognition.

Initially, all weights of a SOM network are initialized with uniformly random values in the range  $[0, 1]$ . Once an input pattern  $\vec{X}$  has been applied to the network, it is operated by each neuron in the ‘cortex’ (i.e. the inner prod-

uct between the input signal and weight is calculated for every neuron), and the neuron yielding the maximum output is identified. Observe that this neuron corresponds to that with weights most similar to the input pattern. The above learning rule is then applied not only to the maximum output neuron, to further reinforce its similarity with the input pattern, but also to its neighbors within a circle of radius  $R$  along the ‘cortex’, as illustrated in Figure 10.

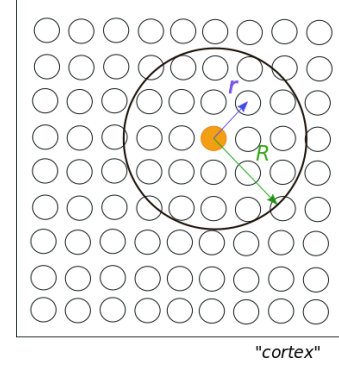


Figure 10: (a) Given an input pattern  $\vec{X}$  presented to a SOM during the learning stage, all neurons within at distance up to  $R$  from the maximum output cell along the cortex have their weights modified in order to become more similar to the input signal.

The learning rate  $\alpha$  is often made to vary with the distance  $r$  and number of training steps  $t$  (e.g. number of patterns trained so far). Generally,  $\alpha$  can be made to decrease exponentially with  $r$ , implementing the tendency of neurons further away from the maximum output neuron to be less affected than those that are closer to that cell. Also,  $\alpha$  is often made to decrease along the training steps  $t$ , in order to contribute to the stability of the unfolding weight modifications. A possible means to achieve this is by expressing  $\alpha$  as:

$$\alpha = \alpha(r, t) = \gamma \exp \left\{ -\frac{(c)(r)(t)}{N_t} \right\} \quad (6)$$

where  $c$  is a constant (e.g. 0.5) controlling the intensity of the exponential decay,  $\gamma$  is a constant defining the maximum learning rate, and  $N_t$  is the total number of patterns to be trained. Figure 11 illustrates  $\alpha(r, t)$  in terms of  $r$  and  $t$  for a specific parameter configuration.

Observe that, though the SOM neurons are not directly interconnected, they nevertheless influence one another during the learning stage as a consequence of the above described modifications not only of the maximum output cell, but also of its respective neighbors.

The basic steps of the described learning stage of a SOM network are summarized in the pseudo-algorithm 1.

Once a SOM network has been trained, it will tend to present adjacent regions associated to each of the types



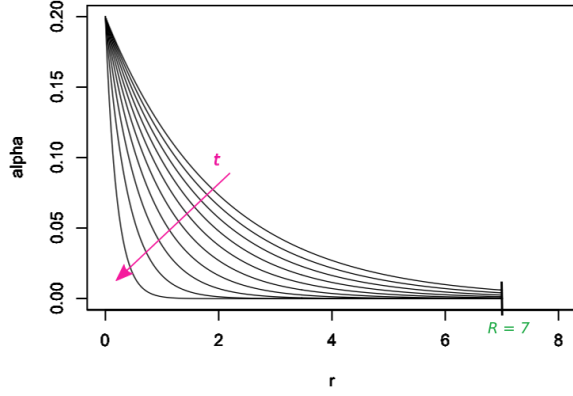


Figure 11: The learning rate values considered in terms of the distance  $r$  from the maximum output cell, which are progressively reduced along the training steps  $t$ . In this figure, we adopted  $R = 7$ ,  $c = 0.5$  and  $\gamma = 0.2$ .

---

#### Algorithm 1 *SOM*

---

1. initialize the SOM network weight with values uniformly distributed in the interval  $[0, 1]$ .
  2. for each pattern  $\vec{X}$  to be trained:
    - (a) input  $\vec{X}$  into the SOM;
    - (b) find cortical cell  $C_{max}$  with the maximum output;
    - (c) apply the learning rule to the weights of  $C_{max}$  and its neighbors;
  3. Output weights.
- 

of trained patterns. Neurons inside each of these regions will tend to present high output when signals resembling the associated pattern is fed into the network. A possible means to identify the patterns to each obtained region is by comparing (e.g. by taking the Euclidean distance) the respective weights of each neuron with the patterns and

taking the pattern that gives the smallest distance.

Now, we present a complete example of training a SOM, taking into account the three basic patterns shown in Figure 12 (a-c). Scrambled versions of these patterns, obtained by shifting the pixels, as illustrated in Figure 12(d-f), are also used during the training stage. A total of 2000 such input patterns were fed into the SOM network during its training.

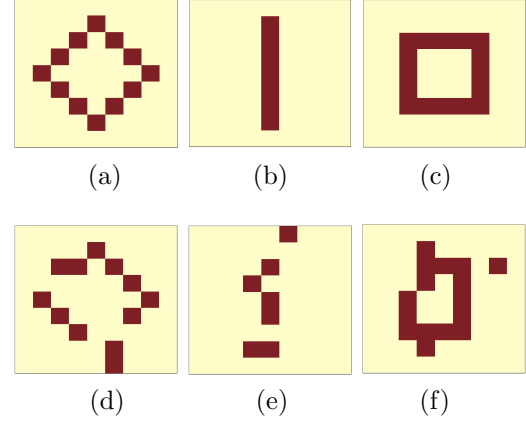


Figure 12: The three reference patterns (a-c) and respective examples of scrambled variations (d-f) used during the described SOM example.

The resulting weights are assigned to patterns by considering their minimal Euclidian distance. Figure 13 (center) depicts the resulting ‘cortex’, already associated to each of the three considered types of patterns. Observe that the cortical space is partitioned among three regions corresponding to each of the types of patterns. A possible trajectory along the ‘cortex’ is indicated in cyan, performing a loop while going through each of the three obtained regions, and some of the respective patterns (defined by the associated weights) found along this trajectory are also shown around the ‘cortex’ figure.

Interestingly, the patterns found along such trajectories tend to vary smooth and progressively, in a way similar to pattern morphing. This allows generalization of the SOM to several patterns that have not been trained, as well as caters for some level of robustness derived from some level of redundancy in the achieved pattern representation. For instance, in case some of the neurons fail (e.g. in a respectively modeled biological network), the neighboring cells would still be able to provide some level of pattern identification.

Once the SOM is trained and associated to the respective pattern types, new patterns can be fed into it, being recognized as a possible instance of the type of pattern corresponding to the neuron yielding the maximum output.



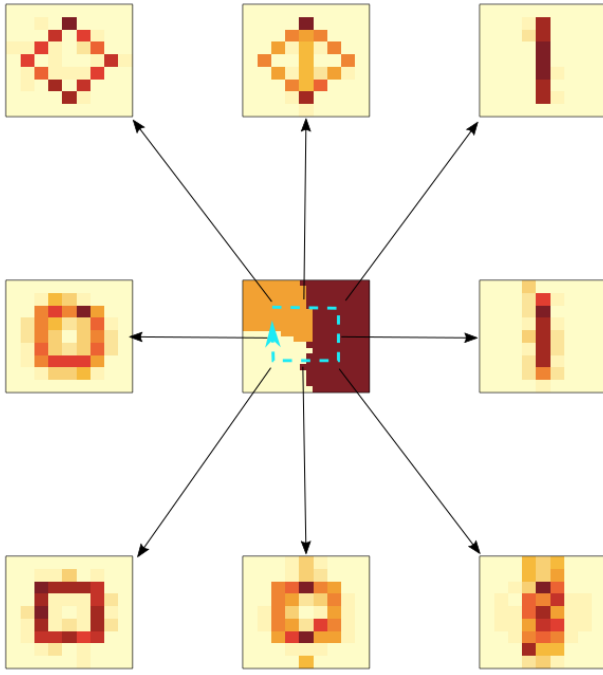


Figure 13: The self-organized map ( $20 \times 20$  cortex at the center of the figure) obtained with respect to instances of the  $9 \times 9$  patterns  $\diamond$ ,  $|$ , and  $\square$ , containing three respective regions marked by distinct colors. The weights of some of the cortical neurons indicated by arrows are also shown surrounding the cortex, illustrating their similarity with variations of the trained patterns, which account for generalization and redundancy in during recognition. Observe that the sequence of weights that ones obtain while moving along the cortical cells (e.g. cyan arrow) can be understood as a kind of pattern morphing.

## 8 Concluding Remarks

Neurons, the ‘atoms’ of the nervous system, can often be understood as performing pattern recognition. In this work, we discussed two possible such interpretations as: (a) linear discrimination, and (b) template matching. We have seen that the output of neurons as in the adopted mathematic-computational model perform the inner product between the input pattern and their respective weights, therefore providing an indication about the similarity between these two vectors. As the inner product is an intrinsic component of convolution, the systematic application of neurons to signals along a given domain becomes related to convolution, being therefore related to convolutional artificial neuronal networks.

After describing the simple and elegant principle of linear morphing of patterns, this concept as well as the interpretation of neurons as template matchers were considered while presenting self-organizing maps, more specifically a basic Kohonen network, also illustrating that trajectories along the respective ‘cortex’ tend to be related to the morphing between the trained patterns.

## Acknowledgments.

Luciano da F. Costa thanks CNPq (grant no. 307085/2018-0) for sponsorship. This work has benefited from FAPESP grant 15/22308-2.

## References

- [1] L. da F. Costa. Modeling: The human approach to science. Researchgate, 2019. [https://www.researchgate.net/publication/333389500\\_Modeling\\_The\\_Human\\_Approach\\_to\\_Science\\_CDT-8](https://www.researchgate.net/publication/333389500_Modeling_The_Human_Approach_to_Science_CDT-8). Online; accessed 03-June-2019.
- [2] H. F. de Arruda, A. Benatti, C. H. Comin, and L. da F. Costa. Learning deep learning. Researchgate, 2019. [https://www.researchgate.net/publication/335798012\\_Learning\\_Deep\\_Learning\\_CDT-15](https://www.researchgate.net/publication/335798012_Learning_Deep_Learning_CDT-15). Online; accessed 22-Dec-2019.
- [3] E. R. Kandel, J. H. Schwartz, T. M. Jessel, S. A. Siegelbaum, and A. J. Hundspeth. *Principles of Neural Science*. McGraw-Hill, 2009.
- [4] S. Haykin. *Neural Networks And Learning Machines*. Pearson, 3rd edition, 2018.
- [5] L. da F. Costa. Convolution! Researchgate, 2019. [https://www.researchgate.net/publication/336601899\\_Convolution\\_CDT-14](https://www.researchgate.net/publication/336601899_Convolution_CDT-14). Online; accessed 09-March-2020.
- [6] T. Kohonen. *Self-Organization and Associative Memory*. Springer Verlag, 2009.
- [7] E. B. Goldstein. *Sensation and Perception*. Cengage Learning, 2012.

CDTs intend to be a halfway point between a formal scientific article and a dissemination text in the sense that they: (i) explain and illustrate concepts in a more informal, graphical and accessible way than the typical scientific article; and (ii) provide more in-depth mathematical developments than a more traditional dissemination work.

It is hoped that CDTs can also incorporate new insights and analogies concerning the reported concepts and methods. We hope these characteristics will contribute to making CDTs interesting both to beginners as well as to more senior researchers.

Each CDT focuses on a limited set of interrelated concepts. Though attempting to be relatively self-contained, CDTs also aim at being relatively short. Links to related material are provided in order to complement the covered subjects.

Observe that CDTs, which come with absolutely no warranty, are non distributable and for non-commercial use only.

The complete set of CDTs can be found at: <https://www.researchgate.net/project/Costas-Didactic-Texts-CDTs>.