Computation with phase oscillators: An oscillatory perceptron model

Pablo Kaluza*

Abteilung Physikalische Chemie, Fritz-Haber-Institut der Max-Planck-Gesellschaft, Faradayweg 4-6, 14195 Berlin, Germany

ARTICLE INFO

ABSTRACT

We present a perceptron model with processing units consisting of coupled phase oscillators. The processing units are able to compute the input signals through a high order synapse mechanism. We show how a network of these elements can be used in analogy to the classical multilayer feedforward neural network. The main characteristics of the classical multilayer perceptron model are conserved, as for example, the backpropagation algorithm for learning. This model of coupled phase oscillators can be seen as a generic study in order to use different kind of oscillators for computational tasks.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The perceptron is one of the first models of artificial neural networks, inspired in biological neural systems, widely accepted [1]. Its use in multilayer feedforward networks with the backpropagation algorithm for learning is a paradigmatic model in this field [2]. Although this model has been quite well understood for theoretical studies and applications, it has strong simplifications as artificial dynamics (updating schemes) and simple node operation. On the other hand, realistic models considering spiking neural networks also have been developed, allowing the use of a modified backpropagation algorithm for learning [3].

Phase oscillator models can be seen as a middle level of description between simple and realistic models. The importance of this intermediate level consists on the generality of the results. In effect, a phase oscillator is a generic description for a wide variety of oscillators on the onset of their limit cycle behavior. On the other hand, its low level allows us to apply, directly, the results of simple models and powerful algorithms.

As a matter of example, phase oscillator networks based on the Kuramoto model have shown to be a good first approximation for oscillatory artificial neural networks and they have been used to perform some kind of computational operations [4]. Hopfield-like models of associative memory [5] have been developed using phase oscillator networks (Aoyagi [6], Aonishi [7] and Nishikawa [8]). Additionally, a phase oscillator model has been used in the construction of a central pattern generator by using the symmetry breaking of a pacemaker, allowing to the system evolves between its fixed points [9].

The aim of this work is to combine oscillatory models of nodes with the results of classical feedforward neural networks. This combination is a step forward in order to extent the utility of powerful computational algorithms to more realistic models of neural systems. Our approach is to use processing units of phase oscillators in multilayer feedforward networks. In order to use in this way these oscillators, we encode their activities as phase differences between them. As result we show that a phase oscillator perceptron behaves in analogy to the classical perceptron model, important fact that allows us to construct feedforward networks, and use the backpropagation algorithm for learning.

This work is organized as following. We present in the second section a model of interaction between phase oscillators, the phase oscillator perceptron model and the feedforward networks. In the section numerical study, we show a network able to solve the XOR logic problem. In the fourth section, we show a linear stability analysis for this model. Finally, we present a discussion and the conclusions in the last section.

2. The model

Consider the case of two phase oscillators, a processing oscillator ϕ_P and a reference oscillator ϕ_R ($\phi_i \in [0 : 2\pi)$, with $i = P, R$). We design a dynamics for them where its fixed points are located in the phase difference relationships $\Delta\phi_{PR} = \pi$ and $\Delta\phi_{PR} = 0$ (with $\Delta\phi_{PR} = \phi_R - \phi_P$). We get this by using the following energy function:

$$L(\phi_P, \phi_R) = 2f_{PR}\cos(\phi_R - \phi_P) - \frac{1}{2}\cos2(\phi_R - \phi_P). \tag{1}$$

* Tel.: +49 30 84135104; fax: +49 30 84135106.
E-mail address: pkaluza@mendoza-conicet.gob.ar

In this expression, $f_{PR} = f_{RP}$ is the coupling that controls the interaction between the oscillators. When $f_{PR} = 0$, $L(\Delta\phi_{PR})$ has two minima for $\Delta\phi_{PR} = 0$ and $\Delta\phi_{PR} = \pi$, and, a maximum exists between these two points. This case corresponds to a Kuramoto model with twice the periodicity.

Eq. (1) allows us to control the nature of the extreme points (maxima or minima) by tunning the coupling f_{PR}. In order to have $\Delta\phi_{PR} = 0$ the only minimum of the system (unique fixed point of the dynamics) we have to use $f_{PR} \leq -1$. In case that the only minimum should be $\Delta\phi_{PR} = \pi$, we must set $f_{PR} \geq 1$.

The dynamics of the phase oscillators is given by the negative gradient of Eq. (1) as

$$\dot{\phi}_i = \sin 2(\phi_j - \phi_i) - 2f_{ij}\sin(\phi_j - \phi_i) \qquad (2)$$

with $i = P,R$, $j = P,R$ and $i \neq j$. The dynamics of this system corresponds to a phase-locked motion of the oscillators. Eqs. (1) and (2) show clearly that the proposed energy function and dynamics have a second order Fourier term. It is well known that this term in systems of phase oscillators can fall in cases of cluster partitions [10–12]. As a result we have a system of oscillator in which we can control the localization of the stable fixed point of the dynamics as a function of the coupling f_{PR}.

2.1. Phase oscillator perceptron

A phase oscillator perceptron (processing unit) consists on two phase oscillators ϕ_R and ϕ_P as Fig. 1a shows. They are coupled
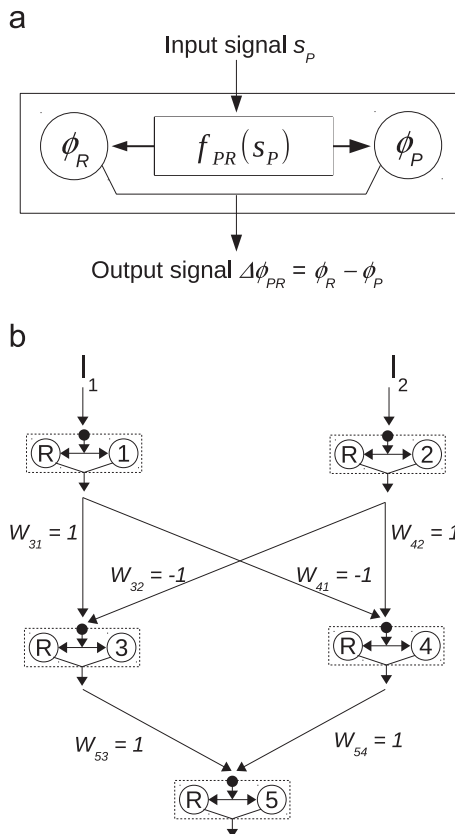
Fig. 1. (a) An oscillatory perceptron (processing unit). It consists of two oscillators, the reference ϕ_R and the processing ϕ_P ones. They are connected by a symmetric synapse $f_{PR}(s_P)$ which is function of the input signal s_P. The unit computes the external signal by this high order synapse mechanism and the result (output signal) is the phase difference between the oscillators and (b) network able to solve the XOR problem. All the nodes have the same reference oscillator ϕ_R (we repeat it in order to simplify the scheme). The thresholds $h_i = 0.5$. The network computes the vector (I_1, I_2) of external signals and returns the value ϕ_5. The XOR problem consists on: $(0,0) \rightarrow 0$, $(0,1) \rightarrow \pi$, $(1,0) \rightarrow \pi$ and $(1,1) \rightarrow 0$.

according with the previous model (Eq. (2)) by a symmetric synapse f_{PR}. In order to process information with this unit we have to define firstly the meaning of activation. In the classical perceptron model, a unit is activated when its output is one, and when it is zero, the perceptron is inactivated. In analogy, we say that a processing unit is activated if the phase difference between its oscillators is π, and inactivated, if this phase difference is zero.

Consider now an external signal s_P arriving to the processing unit. We can compute such signal by making the synapse f_{PR} a function of it as follows:

$$f_{PR}(s_P) = \alpha F(s_P - h_P). \qquad (3)$$

In this expression $F(x)$ is the activation function ($F(x) = 1$ if $x > 0$, and $F(x) = -1$ if $x < 0$), h_P the threshold of the processing unit, and α a constant value. In general we use in this work $\alpha = 2$. It is clear that if $s_P > h_P$, $f_{PR} = \alpha$ and the stable configuration corresponds to a phase difference of π between the oscillators. This case falls in the activation of the unit. If $s_P < h_P$, $f_{PR} = -\alpha$ and the unit is inactivated since the stable configuration corresponds to the configuration with zero phase difference. This system of phase oscillators behaves in analogy to the classical perceptron model.

The architecture of the proposed oscillatory perceptron corresponds to an high order synapses mechanism. In effect, the synapse f_{PR} between the two oscillators is a function of the input signal s_P, that in general as we shall see later, it is the sum of the synapses coming from other nodes of the system. This way of computation is quite different from previous models of phase oscillator networks (Kuramoto-like models) where the synapses or couplings between nodes are constants. High order synapses are well known in biological systems and they have been also considered in artificial neural networks as for example references [13,14] show.

Despite this work does not intent to claim any biological interpretation of the model, it is important to note that the effect of this proposed mechanism is to change the character of the synapse f_{PR} between excitation and inhibition. Although, we propose this mechanism just as an effective process, evidence of this possibility is known in real biological systems as reference [15] presents. Finally, a similar effect can be reached by cotransmission [16].

2.2. Multilayer feedforward perceptron network

We construct a multilayer feedforward network with the previous processing units (nodes). In general, a network has N nodes, and since all these elements have the same reference oscillator, the network has the same number of oscillators ϕ_i with $i = 1,\ldots,N$. The processing units are arrange in k layers with n_k nodes in each one. The input layer $k = 1$ processes the input signals I_l with $l = 1,..,n_1$. Nodes in the output layer return the final result of the computational process. The network architecture is completed by a set of synapses with weights w_{ij} between the nodes, and thresholds h_i for each processing unit. An example of these networks is shown in Fig. 1b.

At this point it is important to comment the role of the reference oscillator ϕ_R. It is clear that this oscillator has fundamental importance in the model since the synapses $f_{iR}(s_i)$ compute the input signals. However, from a mathematical point of view, the existence of the reference is unnecessary. Thus, we take this reference as a constant $\phi_R = 0$ and we remove it from all the labels in the equations. We refer to an processing unit i just by its processing oscillator ϕ_i. Although we make this simplification in the notation, we must remember the importance of this oscillator and the difference between the synapses f_{iR} inside the processing units and the synapses w_{ij} between these units.

The total input signal $s_i(I_i, \overrightarrow{\phi})$ arriving to the processing unit ϕ_i is the following:

$$s_i(I_i, \overrightarrow{\phi}) = \sum_{j=1}^{N} \frac{w_{ij}}{\pi} m(\phi_j) + I_i. \tag{4}$$

In this expression I_i is the external input signal arriving to the unit i ($I_i \neq 0$ only for input units). The synapse or connection from node j to i is weighted by w_{ij} (if there is not a connection $w_{ij} = 0$). The function $m(\phi)$ returns the smaller phase difference of the oscillator ϕ in the interval $[0 : \pi]$. Finally, the vector $\overrightarrow{\phi}$ has N elements with the phase of the oscillators of the network. The unit ϕ_i processes the total input signal $s_i(I_i, \overrightarrow{\phi})$ according to Eq. (3). Thus, the coupling between this oscillator and the reference is given by

$$f_i(s_i(\overrightarrow{I_i}, \overrightarrow{\phi})) = \alpha F(s_i(\overrightarrow{I_i}, \overrightarrow{\phi})) - h_i). \tag{5}$$

In this equation, the parameter h_i is the threshold of the node ϕ_i.

For this system we design a kind of energy or potential for each layer of the network as follows:

$$L_k(\overrightarrow{I}, \overrightarrow{\phi}) = \sum_{i=1}^{n_k} \left(2 f_i(I_i, \overrightarrow{\phi}) \cos(\phi_i) - \frac{1}{2} \cos 2(\phi_i) \right). \tag{6}$$

In this expression the sum is performed on the n_k preceding nodes of the layer k. The nodes of a layer k have couplings $f_i(I_i, \overrightarrow{\phi})$ which are function only of the states of the nodes in the preceding layer, or the corresponding input signals. Thus, the couplings act as an *external field* that controls the dynamics of the layer k. Fixing the external input signals I's, the oscillators in the first layer can only evolve to phase differences zero or π. The oscillators in the second layer follow the external field produced by the first layer, and in consequence, they also can evolve only to phase differences zero and π. And obviously, this process is repeated downstream through the network.

The dynamics for a network is imposed by considering the negative gradient of the potential of each layer (Eq. (6)) as follows:

$$\dot{\phi}_i = 2 f_i(I_i, \overrightarrow{\phi}) \sin(\phi_i) - \sin 2(\phi_i). \tag{7}$$

Since the couplings $f_i(I_i, \overrightarrow{\phi})$ take constant values $\pm \alpha$, the fixed points of this dynamics are located on the corners of a hypercube of N dimensions in the space of phase difference $\{\phi_i\}$. It has one corner in the origin and side of length π. In particular, as we will see later, this dynamics has only one stable fixed point, all the other fixed points are at least saddle ones. This is satisfied if we take the weights and thresholds from a classical feedforward perceptron network which has learned to solve a specific problem. Fixing the input values I_i to the nodes of the input layer, the system evolves to the corresponding learned stable fixed point independent of the initial conditions.

3. Numerical study

We present in this section a system able to solve the logic XOR problem with the phase oscillatory perceptrons. The network is presented in Fig. 1b. It must process the input signals I_1 and I_2 and returns the result in the node ϕ_5. The weights, thresholds and architecture have been taken from a classical multilayer feedforward network given as example in the book of Rojas [17]. Our hypothesis is that we can use the set of weights $\{w_{ij}\}$ and thresholds $\{h_i\}$ coming from a functional classical multilayer perceptron network. Thus, the backpropagation algorithm can be used to learn the weights in an equivalent classical network and then use these parameters in our model.

We use the dynamics of Eq. (7) adding a stochastic term as follows:

$$\dot{\phi}_i = 2 f_i(I_i, \overrightarrow{\phi}) \sin(\phi_i) - \sin 2(\phi_i) + T \eta_i(t). \tag{8}$$

In this expression T is the noise intensity and $\eta_i(t)$ is a stochastic variable with $\langle \eta_i(t) \rangle = 0$ and $\langle \eta_i(t), \eta_j(t') \rangle = \delta_{ij} \delta(t-t')$. The system has been integrated using a stochastic Euler method with $\Delta t = 0.01$ and noise intensity $T = 0.001$.

Fig. 2 shows the evolutions of the phase differences as function of time. As initial conditions for the oscillators we take random phases with uniform distribution between zero and π. Initially, at $t = 0$ and during 25 time units, we set the input signals as $I_1 = 0$ and $I_2 = 0$. We observe in Fig. 2 how the phase differences evolve to zero, in particular, the output node (Fig. 2e) evolves to $\phi_5 = 0$ solving the XOR problem for these input signals.

At $t = 25$ and during the next 25 time units we set the new input signals as $I_1 = 0$ and $I_2 = 1$. We observe that the phase differences evolve to a new state as it is required by the XOR problem. Now, the output oscillator has a phase difference of $\phi_5 = \pi$ at $t \approx 40$. Note that without noise, the system cannot escape from the previous fixed point that is now unstable. Therefore, in order to process a series of input signals, we need to have stochastic dynamics. We change again the input signals at $t = 50$ ($I_1 = 1$ and $I_2 = 0$) and $t = 75$ ($I_1 = 1$ and $I_2 = 1$), and during 25 times unit each. We observe that the network can follow the input
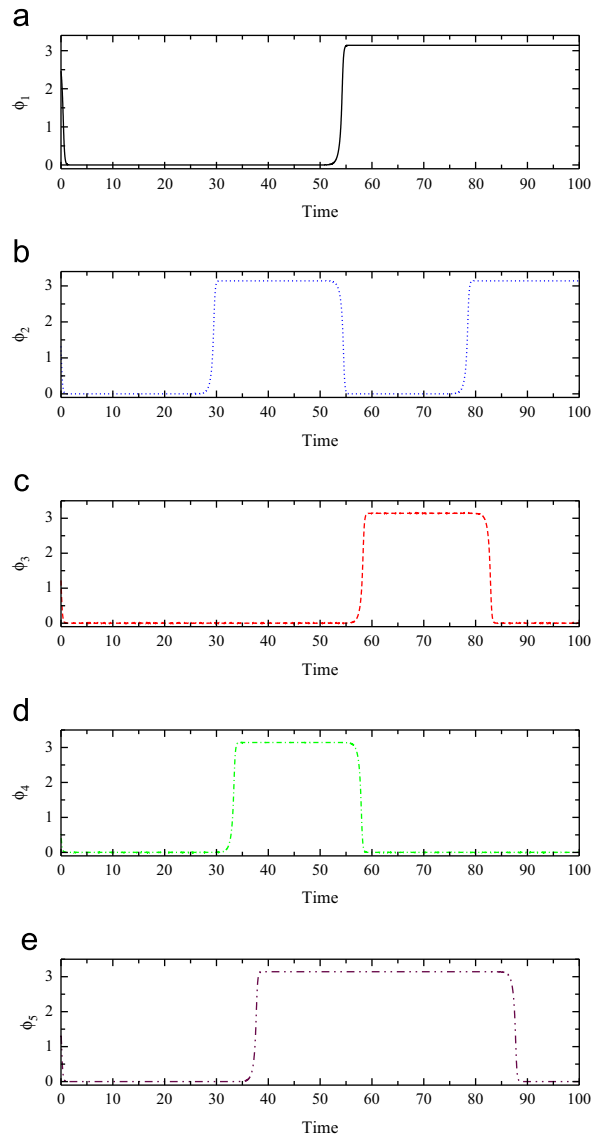

Fig. 2. Phase difference evolutions as function of time of the system shown in Fig. 1b. The input signals change each 25 time units.

signals and process them properly as it is required by the XOR problem ($\phi_5 = \pi$ at $t \approx 70$, and $\phi_5 = 0$ at $t \approx 90$ respectively).

In Fig. 3 we focus in the signaling change at time $t = 25$, from $I_1 = 0$ and $I_2 = 0$, to $I_1 = 0$ and $I_2 = 1$. Fig. 3a shows the layer potentials $L_k(\vec{I}, \vec{\phi})$ as function of time. We observe that when the external signals change at $t = 25$ the potential $L_1(\vec{I}, \vec{\phi})$ of the first layer jumps to a higher value since the actual state of the oscillators in this layer is now located in an unstable fixed point (maximum or saddle point of the potential). Under this situation, the unit ϕ_2 escapes from the unstable fixed point because the noise effect and increases its phase to $\phi_2 = \pi$ (Fig. 3b) and $L_1(\vec{I}, \vec{\phi})$ decreases to the minimum value again.

The second layer of the network is still in a stable configuration until $t \approx 29.4$. At such instant, the first layer is reaching the new configuration and changes $L_2(\vec{I}, \vec{\phi})$ increasing its energy, since the actual configuration of the second layer is now located in a maximum or saddle point of $L_2(\vec{I}, \vec{\phi})$. As in the first layer, the process is repeated here and the fourth unit evolves to $\phi_4 = \pi$ at $t \approx 34$. Finally, the potential $L_3(\vec{I}, \vec{\phi})$ with the output unit becomes in an unstable configuration at $t \approx 33.4$ when the oscillators of the second layer are reaching their stable configuration. At such instant, $L_3(\vec{I}, \vec{\phi})$ increases its value, and by the gradient dynamics, the output node evolves to the new stable configuration $\phi_5 = \pi$ and $L_3(\vec{I}, \vec{\phi})$ gets its minimum value at $t \approx 38$.

The noise helps to escape from the unstable fixed points of the dynamics. Thus, the noise intensity T has been taken very low in order to perform such function. The noise effect, however, consists on breaking down the phase-locked motion of the oscillators. If the noise intensity is low, the system $\vec{\phi}(t)$ evolves very close to the fixed point $\vec{\phi}^*$, allowing the binary information to be properly retrieved. As a result, low noise intensities do not change the stability properties of the system. A more detailed study of the noise effect in systems of phase oscillators can be found in Ref. [11].

With this example we have shown how the system is able to work in analogy with a classical feedforward perceptron network. The network has the ability to process sequentially, and in continuous time, different input signals. Finally, we want to note that the delay between the input signals and the output response (processing time) can be controlled by the noise intensity T and the coupling constants of the dynamics. In effect, high noise intensity T helps to escape faster from the unstable fixed points. On the other hand, the characteristic time of the dynamics of a layer l is controlled by some coupling strength K_l that multiplied the second member of Eq. (7) (we have used $K_l = 1$ in this work). Thus, the response time of each layer can be controlled by tuning the K_l's constants.

4. Stability analysis

We consider the Eq. (7) in the general case of a network with many layers and nodes per layer. Because the feedforward structure of the network, the Jacobian of the system is a triangular matrix, so thus, the diagonal elements of this matrix are the eigenvalues $\{\lambda_i\}$ which determine the linear stability of the system. They are:

$$\lambda_i = \frac{\partial \dot{\phi}_i}{\partial \phi_i} = 2f_i(I_i, \vec{\phi}) \cos(\phi_i) - 2\cos(2\phi_i). \qquad (9)$$

This analysis of stability must be performed taking into account that under perturbations around the fixed points the functions $f_i(I_i, \vec{\phi})$ could change their sign. However, this possibility can be considered annulled by the network construction during the learning. In effect, the learning process does not use a step function as we use in Eq. (3), but a sigmoidal one, as for example, $F(x) = \tanh(\beta x)$ with β in the order of the unity. That activation function during the learning process ensures that the total input signal arriving to a processing unit is relatively far from its threshold value h. In consequence, small perturbations in these networks do not invalidate our linear stability analysis.

As we have seen, the fixed points are located on the corners of the hypercube in the space of phase differences where $\phi_i = 0$ or $\phi_i = \pi$. In a correct configuration over these points the functions $f_i(I_i, \vec{\phi})$ and the phase differences should be compatible. That means that only the pairs: $f_i(I_i, \vec{\phi}) = \alpha$ and $\phi_i = \pi$, and $f_i(I_i, \vec{\phi}) = -\alpha$ and $\phi_i = 0$, are found. So thus, the eigenvalues $\lambda_i = -2(\alpha + 1)$ are always negative for any value of $\alpha > 0$ and the fixed point is stable. In case of an incorrect configuration we have incompatible pairs: $f_i(I_i, \vec{\phi}) = -\alpha$ and $\phi_i = \pi$, and $f_i(I_i, \vec{\phi}) = \alpha$ and $\phi_i = 0$. Thus, the eigenvalues are $\lambda_i = 2(\alpha - 1)$ and the fixed point is unstable for $\alpha > 1$.

By construction, only one corner of the hypercube can have all these pairs compatible and there is only one stable fixed point in the system. All the other fixed points have at least one incompatible pair, and in consequence, at least one unstable direction.

5. Discussion and conclusion

We have shown that a pair of phase oscillators can work in analogy to the classic perceptron model. The new system does not have only a continuous time dynamics, but also it is modeled by oscillatory elements. This characteristic can open the possibility to employ more realistic models of oscillators as processing units in order to construct networks with computational capacity. In effect, phase oscillators can describe complex oscillatory systems of limit cycles when they are coupled by weak interactions. Note that our approach with oscillators is different from Bohtea et al. [3] where information is encoded as time sequences (intervals between spikes). In our model, the phase difference has the meaning of *mean activity* of neurons, as the original perceptron model has.

The model can process sequentially different input signals by switching between the fixed points of the system. The effect of the input signals (*external fields*) is to drive the system by changing the stability of the fixed points, through a processes of symmetry breaking. This computational method has been used to generate sequential patterns in a network of phase oscillators [9], and recently, in a more general way, studied by Neves et al. [18].
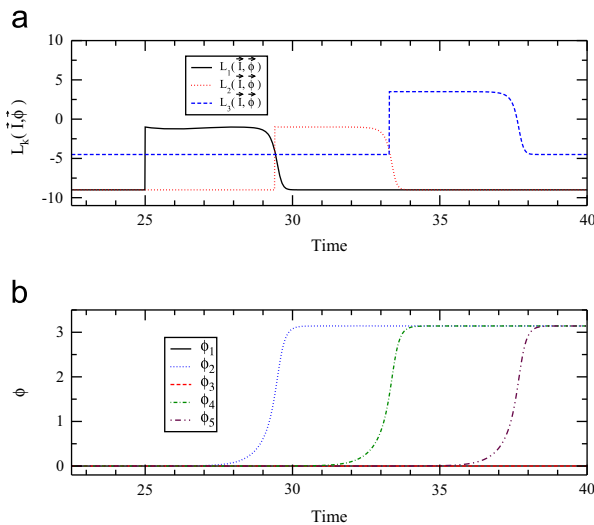

Fig. 3. Layer potentials (a) and phase differences (b) as function of time for the system shown in Fig. 1b. At $t = 25$ the input signals change from $I_1 = 0$ and $I_2 = 0$, to $I_1 = 0$ and $I_2 = 1$.

By using as processing units the phase oscillator perceptrons we have constructed a network able to solve the logic XOR problem. We show that we can use a classical feedforward perceptron network in order to learn the problem, and further, to use the weights and thresholds in our model. Thus, we have a robust system that combines a gradient dynamics for its operation and a very well know algorithm for learning.

The perceptron convergence theorem is satisfied by the presented model. In effect, the processing task done by a classical perceptron unit can be performed by an oscillatory unit by using its weights and threshold. We have to note that the equivalence of the computational properties between a feedforward network of classical perceptrons and one with oscillatory units is ensured only if the activation function is a step function, otherwise, the Kolmogorov's theorem for multilayer networks [19] cannot be immediately applied in our model. The main difference is that using sigmoidal functions the phase oscillator perceptron presents a bistable behavior for $|\alpha F(x)| < 1$ (two stable fixed points at $\phi = 0$ and $\phi = \pi$). Under this situation, the analogy between these two systems breaks down.

Although the presented model is essentially a theoretic one, it can be used to process continuous time input signals in real time. Additionally, the internal dynamics of the network allows to tune the delay between the input signals and the responses on the output layer. Finally, we want to note the possibility of hardware implementation of this model in real physical devices. In effect, devices of electronic oscillators with synchronization properties have been recently proposed for fast sensing and labeling of image objects [20]. Thus, this model is also very interesting from the practical point of view.

References

[1] F. Rosenblatt, Principles of Neurodynamics, Spartan, Washington, DC, 1961.
[2] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.
[3] S.M. Bohtea, J.N. Koka, H. La Poutré, Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing 48 (2002) 17–37.
[4] Y. Kuramoto, Chemical Oscillations, Waves, Turbulence, Springer, New York, 1984.
[5] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. USA 79 (8) (1982) 2554–2558.
[6] T. Aoyagi, Network of neural oscillators for retrieving phase information, Phys. Rev. Lett. 74 (20) (1995) 4075–4078.
[7] T. Aonishi, K. Kurata, M. Okada, Statistical mechanics of an oscillator associative memory with scattered natural frequencies, Phys. Rev. Lett. 82 (13) (1999) 2800–2803.
[8] T. Nishikawa, Y.-C. Lai, F.C. Hoppensteadt, Capacity of oscillatory associative-memory networks with error-free retrieval, Phys. Rev. Lett. 92 (2004) 108101.
[9] P. Kaluza, T. Cioaca, Phase oscillator neural network as artificial central pattern generator for robots, Neurocomputing 97 (2012) 115–124.
[10] I.Z. Kiss, C.G. Rusin, H. Kori, J.L. Hudson, Engineering complex dynamical structures: sequential patterns and desynchronization, Science 316 (2007) 1886–1889.
[11] S.C. Manrubia, A.S. Mikhailov, D.H. Zanette, Emergence of dynamical order: synchronization phenomena in complex systems, in: World Scientific Lecture Notes in Complex Systems, Vol. 2. World Scientific, Singapore, 2004.
[12] D. Hansel, G. Mato, C. Meunier, Clustering and slow switching in globally coupled phase oscillators, Phys. Rev. E 48 (5) (1993) 3470–3477.
[13] P. Peretto, J.J. Niez, Long term memory storage capacity of multiconnected neural networks, Biol. Cybern. 54 (1986) 53–63.
[14] J.J. Arenzon, R.M.C. de Almeida, Neural networks with high-order connections, Phys. Rev. E 48 (1993) 4060–4069.
[15] S.C. Landis, Quick-change artist: from excitatory to inhibitory synapse in minutes, Nat. Neurosci. 5 (2002) 503–504.
[16] G. Burnstock, Cotransmission, Curr. Opin. Pharmacol. 4 (2004) 47–52.
[17] R. Rojas, Neural Networks – A Systematic Introduction, Springer-Verlag, Berlin, New York, 1996.
[18] F.S. Neves, M. Timme, Computation by switching in complex networks of states, Phys. Rev. Lett. 109 (2012) 018701.
[19] V. Kurková, Kolmogorov's theorem and multilayer neural networks, Neural Netw. 5 (1992) 501–506.
[20] J. Kowalski, M. Strzelecki, H. Kim, Implementation of a synchronized oscillator circuit for fast sensing and labeling of image objects, Sensors 11 (4) (2011) 3401–3417.

Pablo Kaluza is a professor of physics at the Basic Science Institute (ICB) of the National University of Cuyo (UNC) in Mendoza Argentina, and, research assistant in the National Scientific and Technical Research Council (CONICET). He holds a bachelor and a master degree in physics from the Balseiro Institute (National University of Cuyo in Bariloche, Argentina). He has obtained a Ph.D. in Physics at the Technical University Berlin in Germany (he has been student at the Fritz Haber Institute of the Max Plank Society in Berlin, Germany). Also, this author has worked as postdoctoral researcher in several German institutions. His actual research focus on dynamical systems, complex networks and artificial neural networks.