A First Model for Hebbian Learning with Spiking Neural P Systems

Miguel A. Gutiérrez-Naranjo, Mario J. Pérez-Jiménez

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012, Sevilla, Spain
E-mails: {magutier,marper}@us.es

Summary. *Spiking neural P systems* and *artificial neural networks* are computational devices which share a biological inspiration based on the transmission of information among neurons. In this paper we present a first model for Hebbian learning in the framework of Spiking Neural P systems by using concepts borrowed from neuroscience and artificial neural network theory.

1 Introduction

> *When an axon of cell A is near enough to excite cell B or repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.*
>
> *D. O. Hebb (1949) [13]*

Neuroscience has been a fruitful research area since the pioneering work of Ramón y Cajal in 1909 [22] and after a century full of results on the man and the mind, many interesting questions are today open problems. Two of such problems of current neuroscience are the understanding of neural plasticity and the neural coding.

The first one, the understanding of neural plasticity, is related to the changes in the amplitude of the postsynaptic response to an incoming action potential. Electrophysiological experiments show that the response amplitude is not fixed over time. Since the 1970's a large body of experimental results on synaptic plasticity has been accumulated. Many of these experiments are inspired by Hebb's postulated (see above). In the integrate-and-fire formal spiking neuron model [9] and also in artificial neural networks [12] is usual to consider a factor w as a measure of the *efficacy* of the synapse from neuron to another.

The second one, the neural coding, is related to the way in which one neuron sends information to other ones. It is interested on the information contained in the spatio-temporal pattern of pulses and on the code used by the neurons to transmit information. This research area wonders how other neurons decode the signal or if the code can be read by external observers and understand the message. At present, a definite answer to these questions is not known.

The elementary processing units in the central nervous system are neurons which are connected to each other in an intricate pattern. Cortical neurons and their connections are packed into a dense network with more than 10^4 cell bodies per cubic millimeter. A single neuron in a vertebrate cortex often connects to more than 10^4 postsynaptic neurons.

The neuronal signals consist of short electrical pulses (also called action potentials or *spikes*) and can be observed by placing a fine electrode close to the soma or axon of a neuron. The junction between two neurons is a *synapse* and it is common to refer to the sending neuron as a presynaptic cell and to the receiving neuron as the postsynaptic cell.

Since all spikes of a given neuron look alike, the form of the action potential does not carry any information. Rather, it is the number and the timing of spikes which matter. Traditionally, it has been thought that most, if not all, of the relevant information was contained in the *mean* firing rate of the neuron. The concept of mean firing rates has been successfully applied during the last 80 years (see, e.g., [18] or [14]) from the pioneering work of Adrian [1, 2]. Nonetheless, more and more experimental evidence has been accumulated during recent years which suggests that a straightforward firing rate concept based on temporal averaging may be too simplistic to describe brain activity. One of the main arguments is that reaction times in behavioral experiment are often too short to allow long temporal averages. Humans can recognize and respond to visual scenes in less than 400ms [24]. Recognition and reaction involve several processing steps from the retinal input to the finger movement at the output. If at each processing steps, neurons had to wait and perform a temporal average in order to read the message of the presynaptic neurons, the reaction time would be much longer. Many other studies show the evidence of precise temporal correlations between pulses of different neurons and stimulus-dependent synchronization of the activity in populations of neurons (see, for example, [5, 11, 10, 6, 23]). Most of these data are inconsistent with a concept of coding by mean firing rates where the exact timing of spikes should play no role.

Instead of considering mean firing rates, we consider the realistic situation in which a neuron abruptly receives an input and for each neuron the timing of the first spike after the reference signal contains all the information about the new stimulus.

Spiking neural P systems (SN P systems, for short) were introduced in [15] with the aim of incorporating in membrane computing[1] ideas specific to spike-based

[1] The foundations of membrane computing can be found in [20] and updated bibliography at [25].

neuron models. The intuitive goal was to have a directed graph were the nodes represent the neurons and the edges represent de synaptic connections among the neurons. The flow of information is carried on the action potentials, which are encoded by objects of the same type, the *spikes*, which is placed inside the neurons and can be sent from presynaptic to postsynaptic neurons according to specific rules and making use of the time as a support of information.

This paper is a first answer to the question proposed by Gh. Păun in [21] related to link the study of SN P systems with neural computing and as he suggests, the starting point has been not only neural computing, but also recent discoveries in neurology.

The paper is organized as follows: first we discuss about SN P systems with input and delay and a new computational device called Hebbian SN P system unit is presented. In section 3 we present our model of learning with SN P systems based on Hebb's postulate. An illustrative experiment carried out with the corresponding software is shown in section 4. Finally, some conclusions and further discussion on some topics of the paper are given in the last section.

2 SN P Systems with Input and Decay

An SN P system consists of a set of neurons placed in the nodes of a directed graph and sending signals (called *spikes*) along the arcs of the graph (called *synapses*). The objects evolve according to a set of rules (called *spiking rules*). The idea is that a neuron containing a certain amount of spikes can consume some of them and produce other ones. The produced spikes are sent (maybe with a delay of some steps) to all neurons to which a synapse exists outgoing from the neuron where the rule was applied. A global clock is assumed and in each time unit each neuron which can use a rule should do it, but only (at most) one rule is used in each neuron. One of the neurons is considered to be the output neuron, and its spikes are also sent to the environment (a detailed description of SN P systems can be found in [21] and the references therein).

In this section we introduce the *Hebbian SN P system unit* which is an SN P system with $m + 1$ neurons (m presynaptic neurons linked to one postsynaptic neuron) endowed with *input* and *decay*. At the starting point all the neurons are inactive. At rest, the membrane of biological neurons has a negative polarization of about $-65mV$, but we will consider the inactivity by considering the the number of spikes inside the neuron is zero. The dynamics of a Hebbian SN P system unit is quite natural. At the starting point, all neurons are at rest and in a certain moment the presynaptic neurons receive spikes enough to activate some rules. The instant of the arrival of the spikes can be different for each presynaptic neuron. These spikes activate one rule inside the neurons and the presynaptic neurons send spikes to the postsynaptic neuron. In the postsynaptic neuron a new rule can be triggered or not, depending on the arrival of spikes and it may send a spike to the environment.

2.1 The Input

The basic idea in SN P systems taken from biological spiking neuron models is the codification of the information in *time*. The information in a Hebbian SN P system unit is also encoded in the time in which the spikes arrive to the neuron and the time in which the new spikes are emitted. The input will be also encoded in time. The idea behind this codification is that the presynaptic neurons may not be activated at the same moment. If we consider a Hebbian SN P system unit as part of a wide neural network, it is quite natural to think that the spikes will not arrive to the presynaptic neurons (and consequently, their rules are not activated) at the same time. In this way, if we consider a Hebbian SN P system unit with m presynaptic neurons $\{u_1, \ldots, u_m\}$, an input will consist of a vector $\vec{x} = \{x_1, \ldots, x_m\}$ of non-negative integers where x_i represents the time unit of the global clock in which the neuron u_i is activated[2].

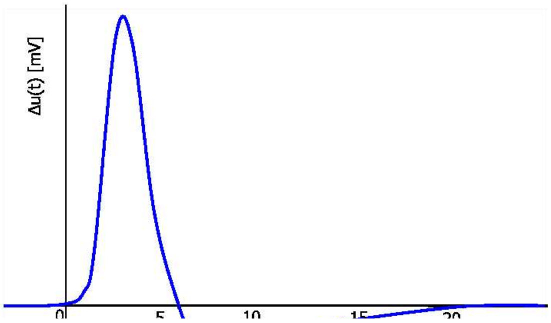
2.2 The Decay

The effect of a spike on the postsynaptic neuron can be recorded with an intracellular electrode which measures the potential difference between the interior of the cell and its surroundings. Without any spike input, the neuron is at rest corresponding to a constant membrane potential. After the arrival of the spike, the potential changes and finally decays back to the resting potential. The spikes, have an amplitude of about 100mV and typically a duration of 1-2 ms. This means that if the total change of the potential due to the arrival of spikes is not enough to activate the postsynaptic neuron, it decays after some milliseconds and the neuron comes back to its resting potential (see Fig. 1).

This biological fact is not implemented in current SN P systems, where the spikes can be inside the neuron for a long time if they are not consumed by any rule. In the Hebbian SN P system unit, we introduce the decay in the action potential of the neurons. When the impulse sent by a presynaptic neuron arrives to the postsynaptic neuron, if it is not consumed for triggering any rule in the postsynaptic neuron it decays and its contribution to the total change of potential in the postsynaptic neuron decreases with time. This decayed potential is still able to contribute to the activation of the postsynaptic rule if other spikes arrive to the neuron and the addition of all the spikes trigger any rule. If this one does not occur, the potential decays and after a short time the neuron reaches the potential at rest. Figure 2 shows a scheme in which two presynaptic neurons send two spikes each of them at different moments to a postsynaptic neuron. Figure 3 shows the changes of potential in the postsynaptic neuron till reaching the threshold for firing a response.

In order to formalize the idea of decay in the framework of SN P systems we introduce a new type of extended rules: the *rules with decay*. They are rules of the form

[2] In Section 5 we discuss about other codings for the input.



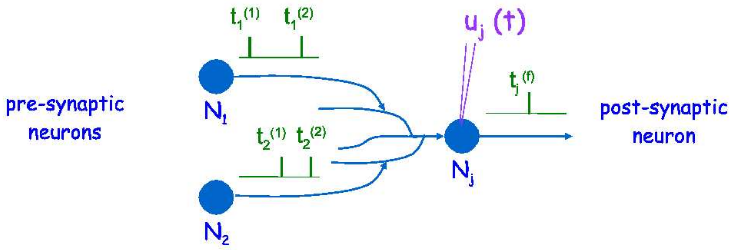


Fig. 2. Two presynaptic and one postsynaptic neuron

$$E/a^k \rightarrow (a^p, S); d$$

where, E is a regular expression over $\{a\}$, k and p are natural numbers with $k \geq p \geq 0$, $d \geq 0$ and $S = (s_1, s_2, \ldots, s_r)$ is a finite non-increasing sequence of natural numbers called the *decaying sequence* where $s_1 = k$ and $s_r = 0$. If $E = a^k$, we will write $a^k \rightarrow (a^p, S); d$ instead of $a^k/a^k \rightarrow (a^p, S); d$.

The intuition behind the *decaying sequence* is the following. When the rule $E/a^k \rightarrow (a^p, S); d$ is triggered at t_0 we look in $S = (s_1, \ldots, s_r)$ for the greatest l such that $p \geq s_l$. Such s_l spikes are sent to the postsynaptic neurons according with the delay d in the usual way. Notice that s_l can be equal to p, so at this point this new type of rule is a generalization of the usual extended rules.

At t_0+d+1, the s_l spikes arrive to the postsynaptic neurons. The decay of such spikes is determined by the decaying sequence. If the spikes are not consumed by the triggering of a rule in the postsynaptic neuron, they decay and at time t_0+d+2 we will consider that $s_l - s_{l+1}$ spikes have disappeared and we only have s_{l+1} spikes in the postsynaptic neuron. If the spikes are not consumed in the following steps by the triggering of a postsynaptic rule, at $t_0 + d + 1 + r - l$ the number of spikes will be decreased to $s_r = 0$ and the spikes are lost.

This definition of decay[3] can be seen as a generalization of the decaying spikes presented in [7]. In that paper a decaying spike a is written in the form (a, e), where $e \geq 1$ is the period. From the moment a spike (a, e) arrives in a neuron, e is decremented by one in each step of computation. As soon as $e = 0$, the corresponding spike is lost and cannot be used anymore.

In this way, a rule $E/a^k \rightarrow a^p; d$ $(k > p)$ where a^p are p decaying spikes (a, e) can be seen with our notation as $E/a^k \rightarrow (a^p, S); d$ with $S = (s_1, \ldots, s_{e+2})$, $s_1 = k$, $s_2 = \cdots = s_{e+1} = p$ and $s_{e+2} = 0$.

2.3 Hebbian SN P System Units

Hebbian SN P system units are SN P systems with a fixed topology endowed with input and decay. They have the following common features:

- The initial number of the spikes inside the neurons is always zero in all Hebbian SN P system units, so we do not refer to them in the description of the unit.
- All the presynaptic neurons are linked to the postsynaptic neuron and these are all the synapses in the SN P system, so they are not provided in the description.
- The output neuron is the postsynaptic one.

Bearing in mind these features, we describe a Hebbian SN P system unit in the following way.

[3] Further discussion about the decay can be found in Section 5.

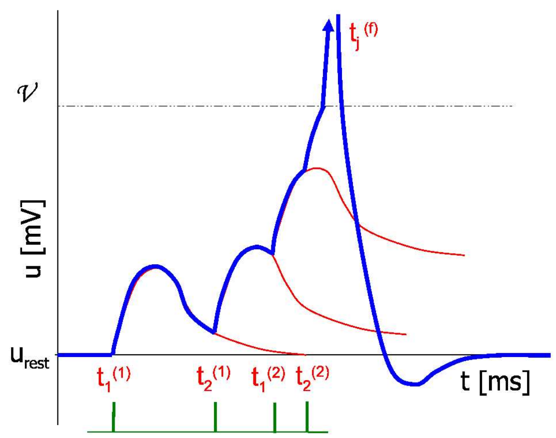


Fig. 3. The potential at the postsynaptic neuron

Definition 1. *A Hebbian SN P system unit of degree m is a construct*

$$H\Pi = (O, u_1, \ldots, u_m, v),$$

where:

- $O = \{a\}$ *is the alphabet (the object a is called spike);*
- $u_1, \ldots, u_m$ *are the presynaptic neurons. Each presynaptic neuron u_i has associated a set of rules $R_i = \{R_{i1}, \ldots, R_{il_i}\}$ where for each $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, l_i\}$, R_{ij} is a decaying rule of the form:*

  $$a^k \to (a^{n_{ij}}, S); d_{ij}$$

  We will call n_{ij} the presynaptic potential *of the rule and d_{ij} is the* delay *of the rule. Note that all rules are triggered by k spikes. The decaying sequence S will be discussed below.*
- *v is the postsynaptic neuron which contains only one postsynaptic rule $E_p^*/a^p \to a; 0$ where E_p^* is the set*[4] *of regular expressions $\{n \in \mathbb{N} \mid n \geq p\}$. We will call p the* threshold *of the postsynaptic potential of the Hebbian SN P system unit.*

By considering the decaying sequences we can distinguish among three types of Hebbian SN P system units:

- Hebbian SN P system units with *uniform decay*. In this case the decaying sequence S is the same for all the rules in the m presynaptic neurons.
- Hebbian SN P system units with *locally uniform decay*. In this case the decaying sequence S is the same for all the rules in each presynaptic neuron.
- Hebbian SN P system units with *non-uniform decay*. In this case each rule has associated a decaying sequence.

A Hebbian SN P system unit is an abstract machine where a global clock is assumed (the system is synchronized). It takes an input and can provide an output or not, depending if the potential in the postsynaptic neuron reaches or not its threshold. The concept of input of a Hebbian SN P system unit is defined as follows:

Definition 2. *An* input *for a* Hebbian SN P system unit *of degree m is a vector $\vec{x} = (x_1, \ldots, x_m)$ of m non-negative integers x_i.*

A Hebbian SN P system unit with input *is a pair $(H\Pi, \vec{x})$ where $H\Pi$ is Hebbian SN P system unit and $\vec{x}$ is an input for it.*

The intuitive idea behind the input is encoding the information in time. Each x_i represent the moment, according to the global clock, in which one spike is provided to each presynaptic neuron.

[4] This rule is an adaptation of the concept of a rule from an extended spiking neural P system with thresholds taken from [7].

2.4 How it works

In this subsection we provide a description of the semantics of a Hebbian SN P system unit. As we saw before, each x_i in the input $\vec{x} = (x_1, \ldots, x_m)$ represents the time in which k spikes are provided to the neuron u_i. At the moment x_i in which the spike arrives to the neuron u_i one rule $(a^k \to (a^{n_{ij}}, S); d_{ij})$ is chosen in a non-deterministic way among all the rules of the neuron.

Applying it means that k spikes are consumed and we look in $S = (s_1, \ldots, s_r)$ for the greatest l such that $n_{ij} \geq s_l$. Such s_l spikes are sent to the postsynaptic neurons according to the delay d_{ij} in the usual way, i.e., s_l spike arrive to the postsynaptic neuron at the moment $x_i + d_{ij} + 1$. The decay of such spikes is determined by the decaying sequence. As we saw above, if the spikes are not consumed by the triggering of a rule in the postsynaptic neuron, they decay and at time $x_i + d_{ij} + 2$ we will consider that $s_l - s_{l+1}$ spikes have disappeared and we only have s_{l+1} spikes in the postsynaptic neuron. If the spikes are not consumed in the following steps by the triggering of a postsynaptic rule, at $x_0 + d_{ij} + 1 + r - l$ the number of spikes will be decreased to $s_r = 0$ and the spikes are lost.

The potential on the postsynaptic neuron depends on the contributions of the chosen rules in the presynaptic neurons. Such rules send spikes that arrive to the postsynaptic neuron at different moments which depend on the input (the moment in which the presynaptic neuron is activated) and the delay of the chosen rule. The contribution of each rule to the postsynaptic neuron also changes along the time due to the decay.

Formally, the potential of the postsynaptic neuron is a natural number calculated as a function R^* which depends on the time t, on the input $\vec{x}$ and on the rules chosen in each neuron $R^*(R_{1i_1}, \ldots, R_{mi_m}, \vec{x}, t) \in \mathbb{N}$. Such a natural number represents the number of the spikes at the moment t in the postsynaptic neurons and it is the result of adding the contributions of the rules $R_{1i_1}, \ldots, R_{mi_m}$.

The Hebbian SN P system unit produces an output if the rule of the postsynaptic neuron v, $E_p^*/a^p \to a$ is triggered, i.e., if at any moment t the amount of spikes in the postsynaptic neuron is greater than or equal to the threshold p, then the rule is activated and triggered. If there does not exist such t, then the Hebbian SN P system unit does not send any spike to the environment.

Bearing in mind the decay of the spikes in the postsynaptic neuron, if any spike has been sent out by the postsynaptic neuron after an appropriate number of steps, any spike will be sent to the environment. From a practical point of view we have a bound for the number of steps in which the spike can be expelled, so we have a decision method to determine if the input $\vec{x}$ provided to the Hebbian SN P system unit produces or not an output.

Example 1. Let us consider the following Hebbian SN P system unit

$$H\Pi = (O, u_1, u_2, v)$$

with non-uniform decay, where:

- $O = \{a\}$ is the alphabet;
- u_1, u_2 are the presynaptic neurons. The presynaptic neurons u_1, u_2 have associated the sets of rules $R_1 = \{R_{11}, R_{12}, R_{13}\}$ and $R_2 = \{R_{21}, R_{22}\}$, respectively, with

$$\begin{array}{ll} R_{11} \equiv a^3 \rightarrow (a^2, (3,2,0)); 0 & R_{21} \equiv a^3 \rightarrow (a^2, (3,2,0)); 1 \\ R_{12} \equiv a^3 \rightarrow (a, (3,1,0)); 1 & R_{22} \equiv a^3 \rightarrow (a, (3,1,0)); 0 \\ R_{13} \equiv a^3 \rightarrow (a^3, (3,0)); 0 & \end{array}$$

- v is the postsynaptic neuron which contains only one postsynaptic rule $E_2^*/a^2 \rightarrow a; 0$.

Notice that in this example, the rules send all the presynaptic potential to the postsynaptic neuron but it only lasts one time unit before being lost. If they are not consumed immediately, they disappear.

Case 1: Let us consider the input $\vec{x} = (0,0)$, i.e., at $t = 0$ three spikes are placed in each presynaptic neuron. We represent the contribution of each rule for $\vec{x} = (0,0)$ in the following table. Notice that for $t \geq 3$ the contribution is zero for all the rules.

	R_{11}	R_{12}	R_{13}	R_{21}	R_{22}
$t=1$	2	0	3	0	1
$t=2$	0	1	0	2	0

Considering the different contributions of the rules and bearing in mind that in each neuron only one rule is non-deterministically chosen, the changes in the postsynaptic potential for $\vec{x} = (0,0)$ are described in the following table.

	$R_{11}\,R_{21}$	$R_{12}\,R_{21}$	$R_{13}\,R_{21}$	$R_{11}\,R_{22}$	$R_{12}\,R_{22}$	$R_{13}\,R_{22}$
$t=1$	**2**	0	**3**	**3**	1	**4**
$t=2$	2	**3**	2	0	1	0

Notice that with the input $\vec{x} = (0,0)$, the postsynaptic neuron activates the rule at $t = 1$ if the chosen rules are $R_{11}\,R_{21}$, $R_{13}\,R_{21}$, $R_{11}\,R_{22}$ or $R_{13}\,R_{22}$. If the chosen rules are $R_{12}\,R_{21}$, then the rule is activated at $t = 2$ and if the chosen rules are $R_{12}\,R_{22}$ then the postsynaptic rule is not activated.

Case 2: Let us consider now the input $\vec{x} = (1,0)$, i.e., at $t = 0$ three spikes are placed in the presynaptic neuron u_2 and in $t = 1$ other three spikes are placed in u_1. As above, we represent the contribution for $\vec{x} = (1,0)$ in the following table.

	R_{11}	R_{12}	R_{13}	R_{21}	R_{22}
$t=1$	0	0	0	0	1
$t=2$	2	0	3	2	0
$t=3$	0	1	0	0	0

The changes of the potential R^* in the postsynaptic potential for $\vec{x} = (1,0)$ are described in the following table.

	$R_{11}\,R_{21}$	$R_{12}\,R_{21}$	$R_{13}\,R_{21}$	$R_{11}\,R_{22}$	$R_{12}\,R_{22}$	$R_{13}\,R_{22}$
$t=1$	0	0	0	1	1	1
$t=2$	**4**	**2**	**5**	**2**	0	**3**
$t=3$	0	1	0	0	1	0

In this case, with the input $\vec{x} = (1, 0)$, the postsynaptic neuron triggers its rule at $t = 2$ but if the chosen rules are $R_{12}\,R_{22}$.

3 Learning

If we look at the Hebbian SN P system units as computational devices where the target is the transmission of information, we can consider that the device *successes* if a spike is sent to the environment and it *fails* if the spike is not sent. In this way, the lack of determinism in the choice of rules is a crucial point in the success of the devices because as we have seen above, if we provide several times the same input, the system can succeed or not.

In order to improve the design of these computational devices and in a narrow analogy with the Hebbian principle, we introduce the concept of *efficacy* in the Hebbian SN P system units. Such efficacy is quantified by endowing each rule with a *weight* that changes along the time, by depending on the contribution of the rule to the success of the device.

According to [8], in Hebbian learning, a synaptic weight is changed by a small amount if presynaptic spike arrival and postsynaptic firing *coincides*. This simultaneity constraint is implemented by considering a parameter s_{ij} which is the difference between the arrival of the contribution of the rule R_{ij} and the postsynaptic firing. Thus, the efficacy of the synapses such that its contributions arrive repeatedly shortly before a postsynaptic spike occurs is increased (see [13] and [3]). The weights of synapses such that their contributions arrive to the postsynaptic neuron *after* the postsynaptic spike is expelled are decreased (see [4] and [16]). This is basically the learning mechanism suggested in [17].

3.1 The Model

In order to implement a learning algorithm in our Hebbian SN P system units, we need to extend it with a set of weights that measure the efficacy of the synapses. The meaning of the weights is quite natural and it fits into the the theory of artificial neural networks [12]: The amount of spikes that arrives to the postsynaptic neuron due to the rule R_{ij} depends on the *contribution* of each rule and also on the *efficacy* of the synapse w_{ij}. As usual in artificial neural networks, the final contribution will be the contribution sent by the rule multiplied by the efficacy w_{ij}.

We fix these concepts in the following definition.

Definition 3. *An extended Hebbian SN P system unit of degree m is a construct*

$$EH\Pi = (H\Pi, w_{11}, \dots, w_{ml_m}),$$

where:

- *$H\Pi$ is a Hebbian SN P system unit of degree m and the rules of the presynaptic neuron u_i are $R_i = \{R_{i1}, \dots, R_{il_i}\}$ with $i \in \{1, \dots, m\}$.*
- *For each rule R_{ij} with $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, l_i\}$, w_{ij} is a real number which denotes the* initial weight *of the rule R_{ij}.*

Associating a weight to each rule means to consider an individual synapse for each rule instead of a synapse associated to the whole neuron. The idea of considering several synapses between two neurons is not new in computational neuron models. For example, in [19] the authors present a model for spatial and temporal pattern analysis via spiking neurons where several synapses are considered. The same idea had previously appeared in [8]. Considering several rules in a neuron and one synapse associated to each rule allows us to design an algorithm for changing the weight (the efficacy) of the synapse according to the result of the different inputs.

The concept of input of a extended Hebbian SN P system unit is similar to the previous one. The information is encoded in time and the input of each neuron denotes the moment in which the neuron is excited.

Definition 4. *An* input *for an* extended Hebbian SN P system unit *of degree m is a vector $\vec{x} = (x_1, \dots, x_m)$ of m non-negative integers x_i.*

An extended Hebbian SN P system unit with input *is a pair $(EH\Pi, \vec{x})$ where $H\Pi$ is an extended Hebbian SN P system unit and $\vec{x}$ is an input for it.*

The semantics

As we saw before, each x_i in the input $\vec{x} = (x_1, \dots, x_m)$ represents the time in which the presynaptic neuron u_i is activated. The formalization of the activation of the neuron in this case differs from the Hebbian SN P system units. The idea behind the formalization is still the same: the postsynaptic neuron receives a little amount of electrical impulse according to the excitation time of the presynaptic neuron and the efficacy of the synapsis. The main difference is that we consider that there exist several synapses between one presynaptic neuron and the postsynaptic one (one synapse for each rule in the neuron) and the potential is transmitted along *all* these synapses according to their efficacy.

Extending the Hebbian SN P system units with efficacy in the synapses and considering that there are electrical flow along all of them can be seen as a generalisation of the Hebbian SN P system units. In Hebbian SN P system units only one rule R_{ij} is chosen in the presynaptic neuron u_i and the contribution emitted by R_{ij} arrives to the postsynaptic neuron according to the decaying sequence. Since the weight w_{ij} multiplies the contribution in order to compute the potential that arrives to the postsynaptic neuron, we can consider the Hebbian SN P system unit as an extended Hebbian SN P system unit with the weight of the chosen rule R_{ij} equals to one and the weight of the remaining rules equals to zero.

At the moment x_i in the presynaptic neuron u_i we will consider that *all* rules $(a^k \to (a^{n_{ij}}, S); d_{ij})$ are activated. The potential on the postsynaptic neuron depends on the contributions of the rules in the presynaptic neurons and the efficacy of the respective synapses. Let us consider that at time x_i the rule $(a^k \to (a^{n_{ij}}, S); d_{ij})$ is activated and the efficacy of its synapse is represented by the weight w_{ij}. When the rule $(a^k \to (a^{n_{ij}}, S); d_{ij})$ is triggered at t_0 we look in $S = (s_1, \ldots, s_r)$ for the greatest l such that $p \times w_{ij} \geq s_l$. Then s_l spikes are sent to the postsynaptic neurons according with the delay d in the usual way.

At t_0+d+1, the s_l spikes arrive to the postsynaptic neurons. The decay of such spikes is determined by the decaying sequence. If the spikes are not consumed by the triggering of a rule in the postsynaptic neuron, they decay and at time t_0+d+2 we will consider that $s_l - s_{l+1}$ spikes have disappeared and we only have s_{l+1} spikes in the postsynaptic neuron. If the spikes are not consumed in the following steps by the triggering of a postsynaptic rule, at step $t_0 + d + 1 + r - l$ the number of spikes will be decreased to $s_r = 0$ and the spikes are lost. The extended Hebbian SN P system unit produces an output if the rule of the postsynaptic neuron v, $E_p^*/a^p \to a$ is triggered.

Bearing in mind the decay of the spikes in the postsynaptic neuron, if the output has not been produced after an appropriate number of steps, no spike will be sent to the environment. From a practical point of view we have a bound for the number of steps in which the spike can be expelled, so we have a decision method to determine if the input $\vec{x}$ provided to the extended Hebbian SN P system unit produces or not an output.

Example 2. Let us consider the extended Hebbian SN P system unit of degree m with uniform decay:

$$H\Pi = (O, u_1, u_2, v, w_{11}, w_{12}, w_{13}, w_{21}, w_{22}),$$

where:

- $O = \{a\}$ is the alphabet;
- u_1, u_2 are the presynaptic neurons. The presynaptic neurons u_1, u_2 have associated sets of rules $R_1 = \{R_{11}, R_{12}, R_{13}\}$ and $R_2 = \{R_{21}, R_{22}\}$, respectively, with

$$\begin{array}{ll} R_{11} \equiv a^{100} \to (a^{40}, S); 0 & R_{21} \equiv a^{100} \to (a^{80}, S); 1 \\ R_{12} \equiv a^{100} \to (a^{70}, S); 1 & R_{22} \equiv a^{100} \to (a^{40}, S); 0 \\ R_{13} \equiv a^{100} \to (a^{30}, S); 0 & \end{array}$$

  The decaying sequence is the same for all the rules, $S = (100, 80, 70, 30, 15, 0)$
- v is the postsynaptic neuron, and it contains only one postsynaptic rule $E_{70}^*/a^{70} \to a; 0$.

- The initial weights are $w_{11} = 0.9$, $w_{12} = 1.2$, $w_{13} = 0.5$, $w_{21} = 0$ and $w_{22} = 1$

In order to compute the function of the postsynaptic potential we need an input. Let us consider $\vec{x} = (1, 0)$. Let us focus on the first rule $R_{11} \equiv a^{100} \to (a^{40}, S); 0$. At $t = 1$ the rule is activated. According to its efficacy, 30 spikes will be placed in the postsynaptic neuron at $t = 2$, since $70 > 40 \times 0.9 = 36 \geq 30$. At $t = 3$ the contribution of this rule is 15 due to the decay and for $t \geq 4$ the contribution is zero. The second rule $R_{12} \equiv a^{100} \to (a^{70}, S); 1$ is also activated at $t = 1$. Due to the delay $d_{12} = 1$, the spikes sent by this rule will be placed at the postsynaptic neuron at $t = 3$. The number of emitted spikes will be 80 since $100 > 70 \times 1.2 = 84 \geq 80$. These spikes will decay in the following steps. We summarize the contributions in the following table. The last column represents the final contribution in the postsynaptic neuron by adding the partial contribution of all the rules.

	R_{11}	R_{12}	R_{13}	R_{21}	R_{22}	$\sum$
$t = 1$	0	0	0	0	30	30
$t = 2$	30	0	15	0	15	60
$t = 3$	15	80	0	0	0	**95**
$t = 4$	0	70	0	0	0	70
$t = 5$	0	30	0	0	0	30
$t = 6$	0	15	0	0	0	15

At time $t = 3$ the postsynaptic potential reaches the value 95 and it is the fist time that it is grater than the threshold, so the postsynaptic rule $E^*_{70}/a^{70} \to a; 0$ is activated and in the next step one spike is sent to the environment.

3.2 The Learning Problem

Let us come back to the Hebbian SN P system units. In such units, provided an input $\vec{x}$, success can be reached or not (i.e., the postsynaptic rule is triggered or not) depending on the non-deterministically rules chosen. In this way, the choice of some rules is *better* than the choice of other ones, by considering that a rule is *better* than another if the choice of the former leads us to the success with a higher probability than the choice of the latter. Our target is to learn which are the best rules according to this criterion.

Formally, a *learning problem* is a 4-uple $(EH\Pi, X, L, \epsilon)$, where:

- $EH\Pi$ is an extended Hebbian SN P system unit
- $X = \{\vec{x_1}, \ldots \vec{x_n}\}$ is a finite set of *inputs* of $EH\Pi$.
- $L : \mathbb{Z} \to \mathbb{Z}$ is a function from the set of integer numbers onto the set of integer numbers. It is called the *learning function*.
- ϵ is a positive constant called the *rate of learning*.

The *output* of a learning problem is an extended Hebbian SN P system unit.

Informal description of the algorithm

Let us consider an extended Hebbian SN P system $EH\Pi$, a learning function $L : \mathbb{Z} \to \mathbb{Z}$ and a rate of learning ϵ.. Let us consider an input $\vec{x}$ and we will denote by $t_{\vec{x}}$ the moment when the postsynaptic neuron reaches the potential for the trigger of the postsynaptic neuron. If such potential is not reached (and the postsynaptic neuron is not triggered) then $t_{\vec{x}} = \infty$.

On the other hand, for each rule $R_{ij} \equiv a^k \to (a^{n_{ij}}, S); d_{ij}$ of a presynaptic neuron we can compute the moment $t^{\vec{x}}_{ij}$ in which its contribution to the postsynaptic potential arrives to the postsynaptic neuron. It depends on the input $\vec{x}$ and the delay d_{ij} of the rule

$$t^{\vec{x}}_{ij} = \vec{x}_i + d_{ij} + 1$$

where $\vec{x}_i$ is the i-th component of $\vec{x}$.

We are interested in the influence of the rule R_{ij} on the triggering of the postsynaptic neuron. For that we need to know the difference between the arrival of the contribution $t^{\vec{x}}_{ij}$ and the moment $t_{\vec{x}}$ in which the postsynaptic neuron is activated.

For each rule R_{ij} and each input $\vec{x}$, such a difference is

$$s^{\vec{x}}_{ij} = t_{\vec{x}} - t^{\vec{x}}_{ij}$$

- If $s^{\vec{x}}_{ij} = 0$, then the postsynaptic neuron reaches the activation exactly in the instant in which the contribution of the rule R_{ij} arrives to the postsynaptic neuron. This fact leads us to consider that the contribution of R_{ij} to the postsynaptic potential has had a big influence on the activation of the postsynaptic neuron.
- If $s^{\vec{x}}_{ij} > 0$ and it is *small*, then the postsynaptic neuron reaches the activation a bit later than the arrival of the contribution of the rule R_{ij} to the postsynaptic neuron. This fact leads us to consider that the contribution of R_{ij} to the postsynaptic potential has influenced on the activation of the postsynaptic neuron due to the decay, but it is not so important as in the previous case.
- If $s^{\vec{x}}_{ij} < 0$ or $s^{\vec{x}}_{ij} > 0$ and it is not *small*, then the contribution of R_{ij} has no influence on the activation of the postsynaptic neuron.

The different interpretations of *small* or *big influence* are determined by the different *learning functions* $L : \mathbb{Z} \to \mathbb{Z}$. For each rule R_{ij} and each input $\vec{x}$, $L(s^{\vec{x}}_{ij}) \in \mathbb{Z}$ measures de degree of influence of the contribution of R_{ij} to the activation of the postsynaptic neuron produced by the input $\vec{x}$.

According to the principle of Hebbian learning, the efficacy of the synapses such that their contributions influence on the activation of the postsynaptic neuron must be increased. The weights of synapses such that their contributions have no influence on the activation of the postsynaptic neuron are decreased.

Formally, given an extended Hebbian SN P system $H\Pi$, a learning function $L : \mathbb{Z} \to \mathbb{Z}$, a rate of learning ϵ and an input $\vec{x}$ of $H\Pi$, the *learning algorithm* outputs a new extended Hebbian SN P system $H\Pi'$ which is equal to $H\Pi$, but the weights: each w_{ij} has been replaced by a new w'_{ij}

$$w'_{ij} = w_{ij} + \epsilon L(s_{ij}^{\vec{x}})$$

Depending on the sign of $L(s_{ij}^{\vec{x}})$, the rule R_{ij} will increase or decrease its efficacy. Note that $L(s_{ij}^{\vec{x}})$ is multiplied by the *rate of learning* ϵ. This rate of learning is usual in learning process in artificial neural networks. It is usually a small number which guarantees that the changes on the efficacy are not abrupt.

Finally, given a *learning problem* $(H\Pi, X, L, \epsilon)$, the learning algorithm takes $\vec{x} \in X$ and outputs $H\Pi'$. In the second step, the learning problem $(H\Pi', X - \{\vec{x}\}, L)$ is considered and we get a new $H\Pi'$. The process finishes when all the inputs has been consumed and the algorithm outputs the last extended SN P system unit.

Example 3. Let us consider the extended Hebbian SN P system unit of degree m with uniform decay:

$$H\Pi = (O, u_1, u_2, v, w_{11}, w_{12}, w_{13}, w_{21}, w_{22}),$$

where:

- $O = \{a\}$ is the alphabet;
- u_1, u_2 are the presynaptic neurons. The presynaptic neurons u_1, u_2 have associated the sets of rules $R_1 = \{R_{11}, R_{12}, R_{13}\}$ and $R_2 = \{R_{21}, R_{22}\}$, respectively, with

$$\begin{array}{ll} R_{11} \equiv a^{100} \to (a^{40}, S); 0 & \quad R_{21} \equiv a^{100} \to (a^{80}, S); 1 \\ R_{12} \equiv a^{100} \to (a^{70}, S); 1 & \quad R_{22} \equiv a^{100} \to (a^{40}, S); 0 \\ R_{13} \equiv a^{100} \to (a^{30}, S); 0 & \end{array}$$

  The decaying sequence is the same for all the rules, $S = (100, 80, 70, 30, 15, 0)$
- v is the postsynaptic neuron which contains only one postsynaptic rule $E_{70}^{*}/a^{70} \to a; 0$.
- The initial weights are $w_{11} = 1.0$, $w_{12} = 1.0$, $w_{13} = 1.0$, $w_{21} = 1.0$ and $w_{22} = 1.0$

Let us consider the learning problem $(EH\Pi, X, L, \epsilon)$, where

- $EH\Pi$ is the extended Hebbian SN P system unit described above.
- $X = \{\vec{x_1}, \vec{x_2}\}$ with $\vec{x_1} = (0, 2)$ and $\vec{x_2} = (0, 0)$.
- L is the learning function $L : \mathbb{Z} \to \mathbb{Z}$

$$L(s) = \begin{cases} 4 & \text{if } s = 0 \\ 2 & \text{if } s = 1 \\ 1 & \text{if } s = 2 \\ -1 & \text{otherwise} \end{cases}$$

- The rate of learning $\epsilon = 0.1$

Step 1: Let us consider the input $\vec{x} = (0, 2)$. The contribution can be summarised in the following table:

	R_{11}	R_{12}	R_{13}	R_{21}	R_{22}	$\sum$
$t = 1$	30	0	30	0	0	60
$t = 2$	15	70	15	0	0	100
$t = 3$	0	30	0	0	30	60
$t = 4$	0	15	0	80	15	110
$t = 5$	0	0	0	70	0	70
$t = 6$	0	0	0	30	0	30
$t = 7$	0	0	0	15	0	15

Therefore, at time $t = 2$ the potential of the postsynaptic neuron reaches a value greater than the threshold 70, then $t_{(0,2)} = 2$. We can compute now the values $t_{ij}^{(0,2)} = x_i + d_{ij} + 1$, $s_{ij}^{(0,2)} = t_{(0,2)} - t_{ij}^{(0,2)}$ and $L(s_{ij}^{(0,2)})$ for every rule R_{ij}.

After computing the values $L(s_{ij}^{(0,2)})$ for every rule R_{ij}, the new weights are calculated as

$$w'_{ij} = w_{ij} + \epsilon\, L(s_{ij}^{(0,2)})$$

These values are summarised in the following table

	$t_{ij}^{(0,2)}$	$s_{ij}^{(0,2)}$	$L(s_{ij}^{(0,2)})$	w_{ij}	w'_{ij}
R_{11}	1	1	2	1	1.2
R_{12}	2	0	4	1	1.4
R_{13}	1	1	2	1	1.2
R_{21}	4	-2	-1	1	0.9
R_{22}	3	-1	-1	1	0.9

Therefore, after this fist step the new weights are $w'_{11} = 1.2$, $w'_{12} = 1.4$, $w'_{13} = 1.2$, $w'_{21} = 0.9$ and $w'_{22} = 0.9$.

Step 2: Let us consider the new extended Hebbian SN P system unit built by replacing the initial weights by the new w'_{ij} and let us consider the second input $\vec{x_2} = (0, 0)$. The contribution can be summarized in the following table.

	R_{11}	R_{12}	R_{13}	R_{21}	R_{22}	$\sum$
$t = 1$	30	0	30	0	30	90
$t = 2$	15	80	15	70	15	195
$t = 3$	0	70	0	30	0	100
$t = 4$	0	30	0	15	0	45
$t = 5$	0	15	0	0	0	15

Therefore, at time $t = 1$ the potential of the postsynaptic neuron reaches a value greater than the threshold 70, then $t_{(0,0)} = 1$. We can compute now the values $t_{ij}^{(0,0)} = x_i + d_{ij} + 1$, $s_{ij}^{(0,0)} = t_{(0,0)} - t_{ij}^{(0,0)}$ and $L(s_{ij}^{(0,0)})$ for every rule R_{ij}.

After computing the values $L(s_{ij}^{(0,0)})$ for every rule R_{ij}, the new weights are calculated as

$$w''_{ij} = w'_{ij} + \epsilon\, L(s_{ij}^{(0,0)})$$

These values are summarized in the following table

	$t_{ij}^{(0,2)}$	$s_{ij}^{(0,2)}$	$L(s_{ij}^{(0,2)})$	w'_{ij}	w''_{ij}
R_{11}	1	0	4	1.2	1.6
R_{12}	2	-1	-1	1.4	1.3
R_{13}	1	0	4	1.2	1.6
R_{21}	2	-1	-1	0.9	0.8
R_{22}	1	0	4	0.9	1.3

Therefore, after this fist step the new weights are $w'_{11} = 1.6$, $w'_{12} = 1.3$, $w'_{13} = 1.6$, $w'_{21} = 0.8$ and $w'_{22} = 1.3$.

The use of weights needs more discussion. The weights are defined as real numbers and membrane computing devices are discrete. If we want to deal with discrete computation in all the steps of the learning process we have to choose the parameters carefully. The following result gives a sufficient constraint for having an integer number of spikes at any moment.

Theorem 1. *Let a be the greatest non-negative integer such that for all presynaptic potential n_{ij} there exists an integer z_{ij} such that $n_{ij} = x_{ij} \times 10^a$.*

Let b be the smallest non-negative integer such that for all initial weight w_{ij} and for the rate of learning ϵ there exist the integers k_{ij} and k such that $w_{ij} = k_{ij} \times 10^b$ and $\epsilon = k \times 10^b$.

If $a - b \geq 0$, then for all presynaptic potential n_{ij} and all the weights w obtained along the learning process, $n_{ij} \times w$ is an integer number.

In other words, if there exists a and b such that all the presynaptic potentials n_{ij} can be expressed as $n_{ij} = x_{ij} \times 10^a$ for an appropriate integer x_{ij} and the initial weights w_{ij} and rate of learning ϵ can be expressed as $w_{ij} = k_{ij} \times 10^b$ and $\epsilon = k \times 10^b$ for appropriate integer numbers k_{ij}, k and $a - b \geq 0$ then for all presynaptic potential n_{ij} and all the weights w obtained along the learning process, $n_{ij} \times w$ is an integer number.

Proof. It suffices to consider the recursive generation of new weights $w_{n+1} = w_n + \epsilon L(s_n)$ and therefore

$$w_{n+1} = w_0 + \epsilon(L(s_0) + \cdots + L(s_n)).$$

If we develop $n_{ij} \times w_{n+1}$ according to the statement of the theorem, we have

$$\begin{aligned} n_{ij} \times w_{n+1} &= x_{ij} \times 10^a \times [k_0 \times 10^{-b} + (k \times 10^{-b}(L(s_0) + \cdots + L(s_n)))] \\ &= 10^{a-b} \times x_{ij} \times [k_0 + k(L(s_0) + \cdots + L(s_n))] \end{aligned}$$

Since $x_{ij} \times [k_0 + k(L(s_0) + \cdots + L(s_n))]$ is an integer number, if $a - b \geq 0$ then $n_{ij} \times w_{n+1}$ is an integer number.

4 An Experiment

Let us consider the Hebbian SN P system

$$H\Pi = (O, u_1, u_2, v)$$

with uniform decay, where:

- $O = \{a\}$ is the alphabet;
- u_1, u_2 are the presynaptic neurons. The presynaptic neurons u_1, u_2 have associated the sets of rules $R_1 = \{R_{11}, R_{12}, R_{13}\}$ and $R_2 = \{R_{21}, R_{22}\}$, respectively, with

$$\begin{array}{ll} R_{11} \equiv a^{3000} \to (a^{3000}, S); 0 & R_{21} \equiv a^{3000} \to a^{1000}; 0 \\ R_{12} \equiv a^{3000} \to (a^{2000}, S); 1 & R_{22} \equiv a^{3000} \to a^{3000}; 3 \\ R_{13} \equiv a^{3000} \to (a^{2000}, S); 7 & \end{array}$$

- The *decaying sequence* is $S = (3000, 2800, 1000, 500, 0)$.
- v is the postsynaptic neuron which contains only one postsynaptic rule $E^*_{1200}/a^{1200} \to a; 0$.

Let $EH\Pi$ be the Hebbian SN P system unit $H\Pi$ extended with the initial weights $w_{11} = 0.5$, $w_{12} = 0.5$, $w_{13} = 0.5$, $w_{21} = 0.5$ and $w_{22} = 0.5$.

Let us consider the learning problem $(EH\Pi, X, L, \epsilon)$ where

- $EH\Pi$ is the extended Hebbian SN P system unit described above,
- X is a set of 200 random inputs (x_i^1, x_i^2) with $1 \leq 1 \leq 200$ and $x_i^1, x_i^2 \in \{0, 1, \ldots, 5\}$
- L is the learning function $L : \mathbb{Z} \to \mathbb{Z}$

$$L(s) = \begin{cases} 3 & \text{if } s = 0 \\ 1 & \text{if } s = 1 \\ -1 & \text{otherwise} \end{cases}$$

- The rate of learning is $\epsilon = 0.001$

We have programmed an appropriate software for dealing with this learning problems. After applying the learning algorithm, we obtain a new extended Hebbian SN P system unit similar to $EH\Pi$ but with the weights

$$w_{11} = 0.754, \quad w_{12} = 0.992, \quad w_{13} = 0.3, \quad w_{21} = 0.454, \quad w_{22} = 0.460$$

Fig 4 shows the evolution of the weights of the synapses.

The learning process shows clearly the differences among the rules.

- The *worst* rule is R_{13}. In a debugging process of the design of an SN P System network that rule should be removed. The value of the weight has decreased along all the learning process. This fact means that the rule has never contributed to the success of the unit and then it can be removed. The reason is

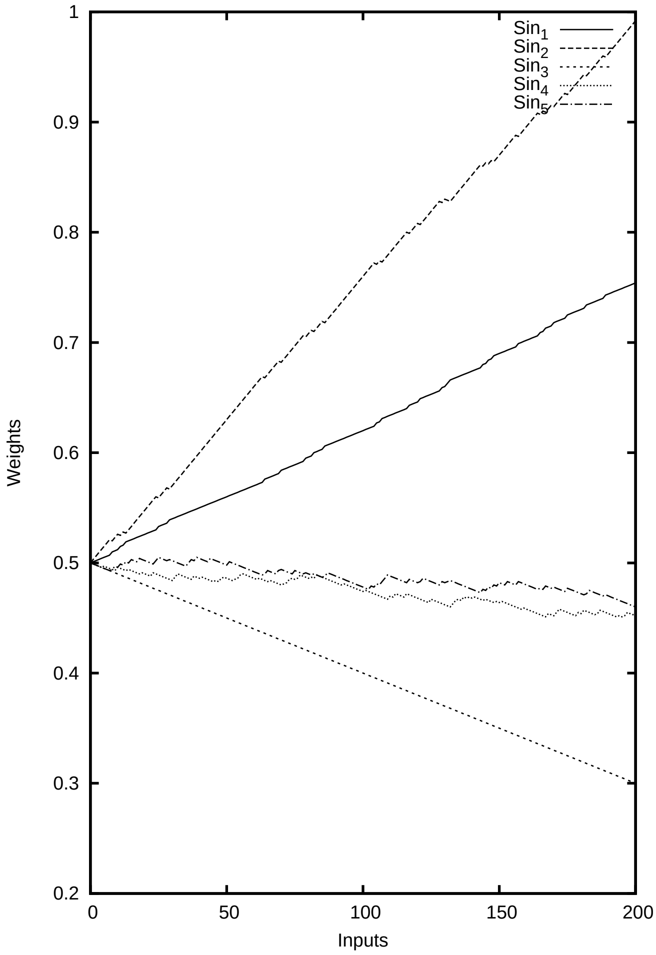


Fig. 4. The evolution of the weights

clear. The rule emits four spikes and the postsynaptic rule is activated with two spikes. Even with the decay, the potential provided by the rule is too much for triggering the rule.

- On the other extreme, the *best* rules are R_{11} and R_{21}. In most of the cases, (not all) these rules have been involved in the success of the unit.
- The other two rules R_{21} and R_{22} have eventually contributed to the success of the unit but not so clearly as R_{11} and R_{21}. We can also guess the reasons. For R_{11}, the presynaptic potential, 1000, has little influence in the postsynaptic potential and for R_{22}, the presynaptic potential is larger than the threshold, but it has a large delay, so the arrival of its potential to the postsynaptic neuron is often later than the activation of the postsynaptic rule.

5 Conclusions and Future Work

The integration in an unique model of concepts from neuroscience, artificial neural networks and spiking neural P systems is not an easy task. Each of the three fields have its own concepts, languages and features. The work of integration consists on choosing ingredients from each field and trying to compose a computational device with the different parts. This means that some of the ingredients used in the devices presented in this paper are not usual in the SN P systems framework. Although the authors have tried to be as close to the SN P system spirit as possible some remarks should be considered.

In the paper, the input of the device is provided as a vector $(t_1, \ldots, t_m)$ of non-negative integers, where t_i represents the moment in which one rule (non-deterministically chosen) of the neuron u_i is activated. Obviously, this is not the usual way to provide the input to an SN P system. Nonetheless, the information encoded in the vector $(t_1, \ldots, t_m)$ can be provided to the input neurons by m spike trains were all the elements are 0's and there is only one 1 in the position t_i. In this way, the input is encoded by m spike trains, which is closer to the standard inputs in SN P systems.

The idea of providing the input with a spike train of 0's and only one 1 in the position t_i carries out new problems. In the literature of SN P systems, in the instant t_i only one spike is supplied to the neuron u_i. In our device we want that a rule of type $a^r \to a^p; d$ is activated with $r > 1$. At this point we can consider several choices. The first one is to consider that at time t_i the spike train provides r spikes, but this choice lead us far from the SN P system theory. A second option is to consider that the spike trains have r consecutive 1's and each of them provide one spike. The remaining elements in the train are zeros. In this way the moment t_i will be the instant in which the r spikes have been provided to the neuron. A drawback for this proposal can be that r can be a big number and this increase the number of steps of the device. A third choice is to consider amplifier modules as in Figure 5. The leftmost neuron receives a spike train where all the elements are 0's but the $t_i - th$ which is 1. At the moment t_i only one spike is supplied to the neuron. At $t_i + 1$, one spike arrives to the r postsynaptic neurons, and each of them sends one spike to the rightmost neuron, so at $t_i + 2$ exactly r spikes arrive simultaneously to the last neuron.

These three solutions can be an alternative to the use of the vector $(t_1, \ldots, t_m)$ and deserve to be considered for further research in this topic.

Another main concept in this paper is the delay. It has strong biological intuition, but it is difficult to insert into the SN P systems theory. The main reason is that if we consider the spike as the information unit it does not make sense to talk about a half of a spike or a third of a spike. In that sense, the approach to decay from [7] is full of sense since one spike exists or it is lost, but its potential it is not decreasing in time.

The key point for the decay in this paper is taken from the definition of *extended* SN P systems. In such devices, a neuron can send a different amount of spikes depending on the chosen rule. So, in such devices the information is not only

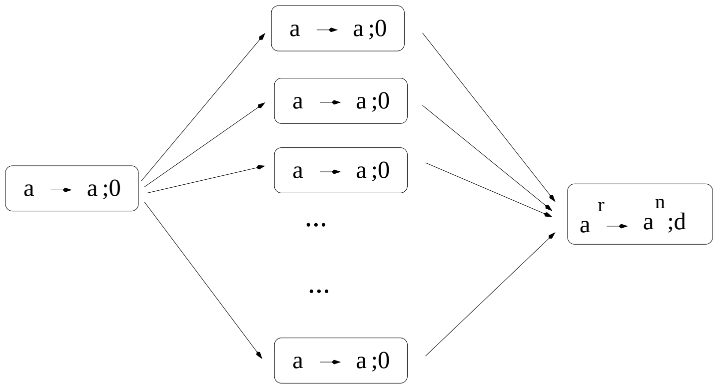


Fig. 5. Amplifier module

encoded in the *time* between two consecutive spikes, but on the *number* of spikes. This lead us to define the decay as a decrement in the number of spikes. In this way, we can consider that a pulse between two neurons is composed by a certain number of spikes which can be partially lost depending on the time.

In this paper, such a decay has been implemented by extending the rules with a finite decreasing sequence which can be uniform, locally-uniform or non uniform for the set of rules. Other implementations are also possible. Probably, the decay can also be implemented with an extra neuron as in Figure 6 which sends to the final neuron a decaying sequence of spikes.

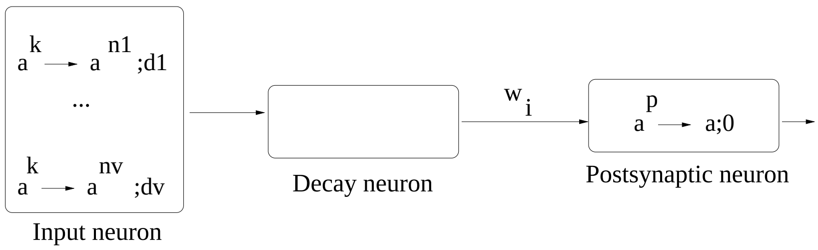


Fig. 6. Including a decay neuron

The use of weights also deserves to be discussed. In Theorem 1 we provide sufficient conditions for handling at every moment an integer number of spikes. In this way, the presented devices keep the principle of discrete computation of SN P systems. Nonetheless, further questions should be considered. For example, the use of negative weights or weights greater than one. Should we consider negative weights and/or a *negative* contribution to the postsynaptic potential? On the other hand, the use weights greater than one leads us to consider that the contribution of one rule to the postsynaptic potential is *greater than* its own presynaptic potential. Can the efficiency of the synapses amplify the potential beyond the number of emitted spikes?

More technical questions are related to the rate of learning and to the algorithm of learning. Both concepts have been directly borrowed from artificial neural networks and need deeper study in order to adapt them to the specific features of SN P systems.

As a final remark, we consider that this paper opens a promising line research bridging SN P systems and artificial neural networks without forgetting the biological inspiration and also opens a door to applications of SN P systems.

Acknowledgements

The authors acknowledge the support of the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the project of excellence TIC-581 of the Junta de Andalucía.

References

1. E.D. Adrian. The impulses produced by sensory nerve endings. J. Physiol. (Lond.), 61, 49-72, 1926.
2. E.D. Adrian. The basis of Sensation. W.W. Norton. New York, 1926.
3. T.V.P. Bliss and G.L. Collingridge. Nature, 361, 31–99, 1993.
4. D. Debanne, B.H. Gähwiler and S.M. Thompson. Proc. Natl. Acsd. Sci. USA, 91, 1148-1152, 1994.
5. R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk and H.J. Reitboeck. Coherent oscillations: A mechanism of feature linking in the visual cortex? Biol. Cybern. 60, 121-130, 1988.
6. A.K. Engel, P. König, A.K. Kreiter, and W. Singer. Interhemispheric sycrhonization of oscillatory neural responses in cat visual cortex. Science, 252, 1177-1179, 1991.
7. R. Freund, M. Ionescu and M. Oswald. Extended spiking neural P systems with decaying spikes and/or total spiking. Proceedings of the International Workshop Automata for Cellular and Molecular Computing, MTA SZTAKI, Budapest, 64-75, 2007.
8. W. Gerstner, R. Kempter, L. van Hemmen and H. Wagner. A neuronal learning rule for sub-millisecond temporal coding. Nature, 383, 76–78, 1996.
9. W. Gerstner and W.Kistler. Spiking Neuron Models. Single Neurons, Populations, Plasticity. Cambridge University Press, 2002.
10. C.M. Gray, P. König, A.K. Engel and W. Singer. Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. Nature, 338, 334-337, 1989.
11. C.M. Gray and W. Singer. Stimulus-specific neuronal oscillations in orientation columns of cat visual cortex. Proc. Natl. Acad. Sci. USA, 86, 1698-1702, 1989.
12. S. Haykin. Neural Networks. A Comprehensive Foundation. Macmillan College Publishin Company, Inc. 1994.
13. D.O. Hebb. The Organization of Behavior, Wiley, New York, 1949.

14. D.H. Hubel and T.N. Wiesel. Receptive Fields of Single Neurons in the Cat's Striate Cortex. J. Phisiol. (Lond.), 148, 574-591, 1959.
15. M. Ionescu, Gh. Păun and T. Yokomori: Spiking neural P systems. *Fundamenta Informaticae*, 71, 2-3, 279-308, 2006.
16. H. Markram and B. Sakmann. Soc. Neurosci. Abstr., 21, 2007, 1995.
17. H. Markram and M. Tsodyks. Redistribution of synaptic efficacy between neocortical pyramidal neurons. Nature, 382, 807–810, 1996.
18. V.B. Mountcastle. Modality anf topographic properties of single neurons of cat's somatosensory cortex. J. Neurophysiol., 20, 408-434, 1957.
19. T. Natschläger and B. Ruf. Spatial and temporal pattern analysis via spiking neurons. Network: Comp. Neural Syst., 9(3), 319-338, 1998.
20. Gh. Păun: *Membrane Computing. An Introduction.* Springer–Verlag, Berlin, 2002.
21. Gh. Păun: Twenty six research topics about spiking neural P systems. In M.A. Gutiérrez-Naranjo, Gh. Păun, A. Romero-Jiménez and A. Riscos-Núñez, editors. Fifth Brainstorming Week on Membrane Computing, Fénix Editora, Sevilla, 263–280, 2007.
22. S. Ramón y Cajal. Histologie du Systeme Nerveux de l'Homme et des Vertébres. A. Maloine, Paris, 1909.
23. W. Singer. The role of synchrony in neocortical processing and synaptic plasticity. In E. Domany, J.L. van Hemmel and K. Schulten, editors, Models of Neural Networks II, chapter 4, Springer Verlag, Berlin, 1994.
24. S. Thorpe, S, Fize and C. Marlot. Speed of processing in the human visual system. Nature, 381, 520-522, 1996.
25. P systems web page `http://ppage.psystems.eu/`