

Hebbian Learning from Spiking Neural P Systems View

Miguel A. Gutiérrez-Naranjo and 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
{magutier,marper}@us.es

Abstract. *Spiking neural P systems and artificial neural networks are computational devices which share a biological inspiration based on the flow 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) [12]

Neuroscience has been a fruitful research area since the pioneering work of Ramón y Cajal in 1909 [20] and after a century full of results on the man and the mind, many interesting questions are still unanswered. 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 postulate (see above). In the integrate-and-fire formal spiking neuron model [8] and also in artificial neural networks [11], it is usual to consider a parameter w as a measure of the *efficacy* of the synapse from a neuron to another.

The second one, the neural coding, is related to the way in which one neuron sends information to other ones and it is interested in 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 whether the code can be read by external observers who can understand the message. At present, a definite answer to these questions is not known.

Since all spikes (short electrical pulses) 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 what matters. 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 decades (see, e.g., [17] or [13]) 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 [21]. 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, [9] or [5]). 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 [14] with the aim of incorporating in membrane computing (more information can be found at [22]) ideas specific to spike-based neuron models. The intuitive goal was to have a directed graph where the nodes represent the neurons and the edges represent the synaptic connections among the neurons. The flow of information is carried by the action potentials, which are encoded by objects of the same type, the *spikes*, which are 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.

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 a corresponding software is shown in Section 4. Finally, some conclusions 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 capable of sending signals (called *spikes*) along the arcs of the graph (called *synapses*) according to specific rules. 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 adjacent neurons 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 distinguished neuron 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 [19] 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 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 enough spikes 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 that the information is encoded 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 $\{u_1, \dots, u_m\}$, an input will consist of a vector $\mathbf{x} = (x_1, \dots, x_m)$ 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 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

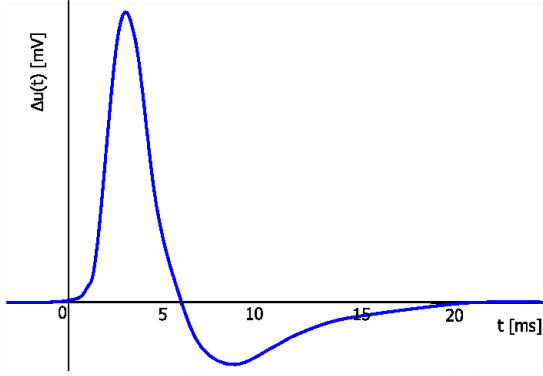


Fig. 1. Dynamics of one spike

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 the changes of potential in the postsynaptic neuron until 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 $E/a^k \rightarrow (a^p, S); d$ where, E is a regular expression over $\{a\}$, k and p are natural numbers with $k \geq p \geq 0$, $d \geq 0$ and $S = (s_1, s_2, \dots, s_r)$ 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 idea 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 $S = (s_1, \dots, s_r)$ for the least 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.

This definition of decay can be seen as a generalization of the decaying spikes presented in [6], where a decaying spike a is written in the form (a, e) , where $e \geq 1$ is the period. From the moment a pair (a, e) arrives to a neuron, e is decremented

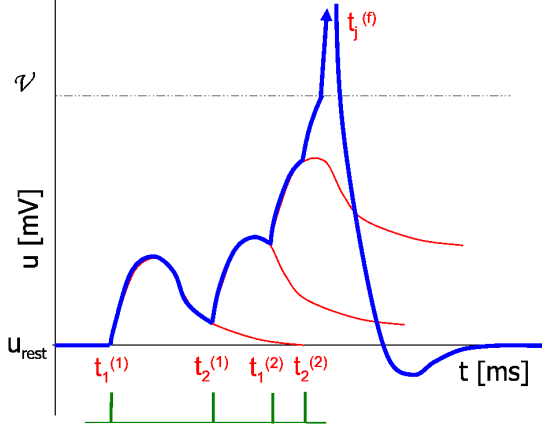


Fig. 2. The potential at the postsynaptic neuron

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 $S = (s_1, \dots, s_{e+2})$, $s_1 = k$, $s_2 = \dots = s_{e+1} = p$ 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 only one 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.

Definition 1. A Hebbian SN P system unit of degree (m, k, p) is a construct $H\Pi = (O, u_1, \dots, u_m, v)$, where:

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

where $k \geq n_{ij} \geq 0$ and $d_{ij} \geq 0$. 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 sequence $S = (s_1, s_2, \dots, s_r)$ is a finite non increasing sequence of natural numbers called the decaying sequence where $s_1 = k$ and $s_r = 0$.

- v is the postsynaptic neuron which only contains¹ the rule $a^p a^* / a^p \rightarrow a; 0$. 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 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.

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

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

The intuitive idea behind the input is encoding the information in time. Each component of the input represents the moment, according to the global clock, in which k spikes are provided to the corresponding presynaptic neuron.

2.4 How It Works

Next we provide a description of the semantics of a Hebbian SN P system unit of degree (m, k, p) . As we saw before, each x_i in the input $\mathbf{x} = (x_1, \dots, x_m)$ represents the time in which k spikes are provided to the neuron u_i . At the moment x_i in which the spikes arrive to the neuron u_i one rule $a^k \rightarrow (a^{n_{ij}}, S); d_{ij}$ is chosen in a non-deterministic way among the rules of the neuron.

Applying it means that k spikes are consumed and we look in $S = (s_1, \dots, s_r)$ for the minimum 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 spikes 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 time $x_0 + d_{ij} + 1 + r - l$ the number of spikes will be decreased to $s_r = 0$ and the

¹ This rule is an adaptation of the concept of a rule from an extended spiking neural P system with thresholds taken from [6].

spikes are lost. Formally, if the chosen rule at the membrane i is $R_{ij} \equiv a^k \rightarrow (a^{n_{ij}}, S); d_{ij}$ with $S = (s_1, \dots, s_r)$ and the rule is activated at time $t = x_i$, then the number of spikes sent by R_{ij} occurring in the postsynaptic neuron at time $t = x_i + d_{ij} + 1 + h$ is s_{k+h} , if $h \in \{0, \dots, r-l\}$ and zero otherwise. The index l is the least index in $\{1, \dots, r\}$ such that $n_{ij} \geq s_l$.

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

The Hebbian SN P system unit produces an output if the rule of the postsynaptic neuron v , $a^p a^* / a^p \rightarrow a$, 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. In fact, we have a lower bound for the number of steps in which the spike can be expelled, so we have a decision method to determine if the input \mathbf{x} produces or not an output².

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 *succeeds* 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 [7], 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

² A detailed description and some examples can be found in [10].

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 [3] and [12]). The weights of synapses such that their contributions arrive to the postsynaptic neuron *after* the postsynaptic spike is expelled are decreased (see [4] and [15]). This is basically the learning mechanism suggested in [16].

3.1 The Model

In order to implement a learning algorithm in our Hebbian SN P system unit, 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 theory of artificial neural networks [11]. 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* w_{ij} of the synapse. 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 $EHP = (HP, w_{11}, \dots, w_{ml_m})$, where:*

- *HP 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 [18] 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 [7]. 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 an 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 extended Hebbian SN P system unit with input is a pair (EHP, \mathbf{x}) , where HP is an Hebbian SN P system unit and \mathbf{x} is an input for it.*

The semantics. As we saw before, each x_i in the input $\mathbf{x} = (x_1, \dots, x_m)$ 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 generalization 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 $a^k \rightarrow (a^{n_{ij}}, S); d_{ij}$ 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 $a^k \rightarrow (a^{n_{ij}}, S); d_{ij}$ is activated and the efficacy of its synapse is represented by the weight w_{ij} . When the rule $a^k \rightarrow (a^{n_{ij}}, S); d_{ij}$ is triggered at the instant t_0 we look in $S = (s_1, \dots, s_r)$ for the least 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 time $t_0 + d + 1$, 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 $v, a^p a^* / a^p \rightarrow a$, is triggered.

3.2 The Learning Problem

Let us come back to the Hebbian SN P system units. In such units, provided an input \mathbf{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 (EHP, X, L, ϵ) , where:

- EHP is an extended Hebbian SN P system unit.
- $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is a finite set of *inputs* of EHP .

- $L : \mathbb{Z} \rightarrow \mathbb{Z}$ 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 $EHII$, a learning function $L : \mathbb{Z} \rightarrow \mathbb{Z}$ and a rate of learning ϵ . Let us consider an input \mathbf{x} and we will denote by $t_{\mathbf{x}}$ the moment when the postsynaptic neuron reaches the potential for the trigger of the postsynaptic neuron. If such potential is not reached then $t_{\mathbf{x}} = \infty$.

On the other hand, for each rule $R_{ij} \equiv a^k \rightarrow (a^{n_{ij}}, S); d_{ij}$ of a presynaptic neuron we can compute the moment $t_{ij}^{\mathbf{x}}$ in which its contribution to the postsynaptic potential arrives to the postsynaptic neuron. It depends on the input \mathbf{x} and the delay d_{ij} of the rule $t_{ij}^{\mathbf{x}} = \mathbf{x}_i + d_{ij} + 1$ where \mathbf{x}_i is the i -th component of \mathbf{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 $t_{ij}^{\mathbf{x}}$ and the moment $t_{\mathbf{x}}$ in which the postsynaptic neuron is activated.

For each rule R_{ij} and each input \mathbf{x} , such a difference is $s_{ij}^{\mathbf{x}} = t_{\mathbf{x}} - t_{ij}^{\mathbf{x}}$

- If $s_{ij}^{\mathbf{x}} = 0$, then the postsynaptic neuron reaches the activation exactly in the instant in which the contribution of the rule R_{ij} arrives to it. 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 $s_{ij}^{\mathbf{x}} > 0$ 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 it. 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 $s_{ij}^{\mathbf{x}} < 0$ or $s_{ij}^{\mathbf{x}} > 0$ 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* $L : \mathbb{Z} \rightarrow \mathbb{Z}$. For each rule R_{ij} and each input \mathbf{x} , $L(s_{ij}^{\mathbf{x}}) \in \mathbb{Z}$ measures the degree of influence of the contribution of R_{ij} to the activation of the postsynaptic neuron produced by the input \mathbf{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 HII , a learning function $L : \mathbb{Z} \rightarrow \mathbb{Z}$, a rate of learning ϵ and an input \mathbf{x} of HII , the *learning algorithm* outputs a new extended Hebbian SN P system HII' which is equal to HII , but the weights: each w_{ij} has been replaced by a new w'_{ij} according to

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

Depending on the sign of $L(s_{ij}^x)$, the rule R_{ij} will increase or decrease its efficacy. Note that $L(s_{ij}^x)$ 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* (HII, X, L, ϵ) , the learning algorithm takes $\mathbf{x} \in X$ and outputs HII' . In the second step, the learning problem $(HII', X - \{\mathbf{x}\}, L)$ is considered and we get a new HII' . The process finishes when all the inputs has been consumed and the algorithm outputs the last extended SN P system unit.

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 x_{ij} such that $n_{ij} = x_{ij} \cdot 10^a$.*

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 $w_{ij} = k_{ij} \cdot 10^{-b}$ and $\epsilon = k \cdot 10^{-b}$.

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

Proof. For the sake of simplicity, we denote by w^r the update weight w after r steps (and w^0 is the initial weight). Then, it suffices to consider the recursive generation of new weights $w^{n+1} = w^n + \epsilon \cdot L(s_n)$, where s_n is the corresponding value in the step n , and therefore

$$w^{n+1} = w^0 + \epsilon \cdot (L(s_0) + \dots + L(s_n)).$$

If we develop $n_{ij} \cdot w^{n+1}$ according to the statement of the theorem, we have that there exist the integers x_{ij} , k_0 and k such that

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

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

4 A Case Study

Let us consider the Hebbian SN P system $HII = (O, u_1, u_2, v)$ 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_i , where $R_1 = \{R_{11}, R_{12}, R_{13}\}$ and $R_2 = \{R_{21}, R_{22}\}$, respectively, with

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

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

Let EHI be the Hebbian SN P system unit HII extended with the initial weights $w_{11} = w_{12} = w_{13} = w_{21} = w_{22} = 0.5$.

Let us consider the learning problem (EHI, X, L, ϵ) where

- EHI 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 $1 \leq i \leq 200$ and $x_i^1, x_i^2 \in \{0, 1, \dots, 5\}$
- L is the learning function $L : \mathbb{Z} \rightarrow \mathbb{Z}$

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

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

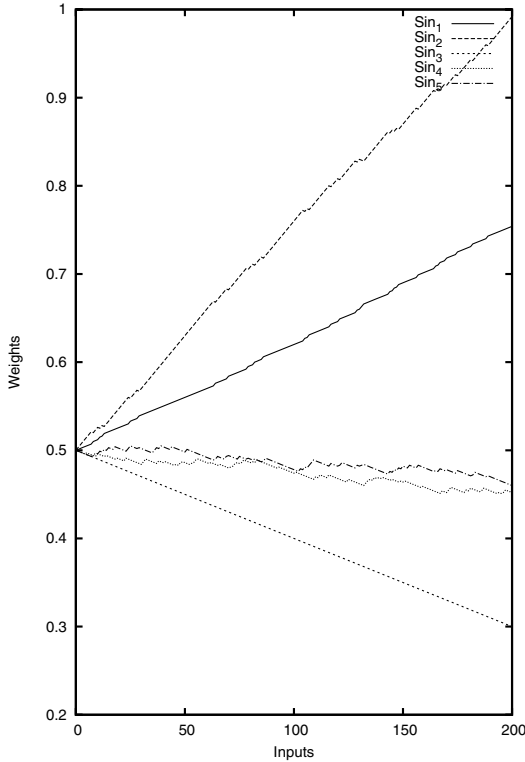


Fig. 3. The evolution of the weights

We have programmed an appropriate software for dealing with learning problems. After applying the learning algorithm, we obtain a new extended Hebbian SN P system unit similar to *EHII* but with the weights $w_{11} = 0.754$, $w_{12} = 0.992$, $w_{13} = 0.3$, $w_{21} = 0.454$, $w_{22} = 0.460$. Figure 3 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 such 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.
- On the contrary, the *best* rules are R_{11} and R_{12} . 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} .

5 Conclusions and Future Work

The integration in a unique model of concepts from neuroscience, artificial neural networks and spiking neural P systems is not an easy task. Each of the three fields has 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, a deeper study related to the input, the decay and the weights is needed.

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 a 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. Adrian, E.D.: The impulses produced by sensory nerve endings: Part I. The Journal of Physiology 61, 49–72 (1926)
2. Adrian, E.D.: The Basis of Sensation. W.W. Norton, New York (1926)

3. Bliss, T.V.P., Collingridge, G.L.: A synaptic model of memory: long-term potentiation in the hippocampus. *Nature* 361, 31–39 (1993)
4. Debanne, D., Gähwiler, B.H., Thompson, S.M.: Asynchronous pre- and postsynaptic activity induces associative long-term depression in area CA1 of the rat hippocampus in vitro. *Proc. National Academy of Sciences* 91, 1148–1152 (1994)
5. Engel, A.K., König, P., Kreiter, A.K., Singer, W.: Interhemispheric synchronization of oscillatory neural responses in cat visual cortex. *Science* 252, 1177–1179 (1991)
6. Freund, R., Ionescu, M., Oswald, M.: Extended spiking neural P systems with decaying spikes and/or total spiking. In: *Proc. Intern. Workshop Automata for Cellular and Molecular Computing, MTA SZTAKI, Budapest*, pp. 64–75 (2007)
7. Gerstner, W., Kempter, R., van Hemmen, L., Wagner, H.: A neuronal learning rule for sub-millisecond temporal coding. *Nature* 383, 76–78 (1996)
8. Gerstner, W., Kistler, W.: *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, Cambridge (2002)
9. Gray, C.M., König, P., Engel, A.K., Singer, W.: Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature* 338, 334–337 (1989)
10. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: A first model for Hebbian learning with spiking neural P systems. In: Díaz-Pernil, D., et al. (eds.) *Sixth Brainstorming Week on Membrane Computing, Fénix Editora*, pp. 211–233 (2008)
11. Haykin, S.: *Neural Networks. A Comprehensive Foundation*. Macmillan College Publishin Company, Inc., Basingstoke (1994)
12. Hebb, D.O.: *The Organization of Behavior*. Wiley, New York (1949)
13. Hubel, D.H., Wiesel, T.N.: Receptive fields of single neurons in the cat's striate cortex. *The Journal of Physiology* 148, 574–591 (1959)
14. Ionescu, M., Păun, G., Yokomori, T.: Spiking neural P systems. *Fundamenta Informaticae* 71, 279–308 (2006)
15. Markram, H., Sakmann, B.: Action potentials propagating back into dendrites triggers changes in efficiency of single-axon synapses between layer V pyramidal neurons. *Soc. Neurosci. Abstr.* 21(1995) (2007)
16. Markram, H., Tsodyks, M.: Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature* 382, 807–810 (1996)
17. Mountcastle, V.B.: Modality and topographic properties of single neurons of cat's somatosensory cortex. *Journal of Neurophysiology* 20(1957), 408–434
18. Natschläger, T., Ruf, B.: Spatial and temporal pattern analysis via spiking neurons. *Network: Computation in Neural Systems* 9, 319–338 (1998)
19. Păun, G.: Twenty six research topics about spiking neural P systems. In: Gutiérrez-Naranjo, M.A., et al. (eds.) *Fifth Brainstorming Week on Membrane Computing, Fénix Editora, Sevilla*, pp. 263–280 (2007)
20. Ramón y Cajal, S.: *Histologie du Systeme Nerveux de l'Homme et des Vertebre*. A. Maloine, Paris (1909)
21. Thorpe, S., Fize, S., Marlot, C.: Speed of processing in the human visual system. *Nature* 381, 520–522 (1996)
22. P systems web page, <http://ppage.psystems.eu/>