IDIAP RESEARCH REPORT

A SUPERVISED LEARNING APPROACH BASED ON STDP AND POLYCHRONIZATION IN SPIKING NEURON NETWORKS

Hélène Paugam-Moisy [1,2], Régis Martinez [1]
and Samy Bengio [2]

IDIAP–RR 06-54

SEPTEMBER 2006

[1] ISC-CNRS, Lyon, France, {hpaugam, rmartinez}@isc.cnrs.fr
[2] IDIAP Research Institute, CP 592, 1920 Martigny, Switzerland and Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, {Helene.Paugam-Moisy, Samy.Bengio}@idiap.ch

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 06-54

A SUPERVISED LEARNING APPROACH BASED ON STDP AND POLYCHRONIZATION IN SPIKING NEURON NETWORKS

Hélène Paugam-Moisy, Régis Martinez
and Samy Bengio

SEPTEMBER 2006

Abstract. We propose a novel network model of spiking neurons, without preimposed topology and driven by STDP (Spike-Time-Dependent Plasticity), a temporal Hebbian unsupervised learning mode, based on biological observations of synaptic plasticity. The model is further driven by a supervised learning algorithm, based on a margin criterion, that has effect on the synaptic delays linking the network to the output neurons, with classification as a goal task. The network processing and the resulting performance are completely explainable by the concept of polychronization, recently introduced by Izhikevich [9]. On the one hand, our model can be viewed as a new machine learning concept for classifying patterns by means of spiking neuron networks. On the other hand, as a model of natural neural networks, it provides a new insight on cell assemblies, a fundamental notion for understanding the cognitive processes underlying memory.
Keywords. Spiking neuron networks, Synaptic plasticity, STDP, Delay learning, Classifier, Cell assemblies, Polychronous groups.

Contents

1 Introduction

Spiking Neuron Networks (SNNs) derive their strength and interest from an accurate modeling of synaptic interactions between neurons, taking into account the time of spike emission. Many biological arguments, as well as theoretical results (e.g. [12, 21, 23]) converge to establish that SNNs are potentially more powerful than traditional artificial neural networks. However, discovering efficient learning rules adapted to SNNs is still a burning topic. At the end of the 90's, solutions were proposed for emulating classic learning rules in SNNs [13, 18, 4], by means of drastic simplifications that often resulted in losing precious features of firing time based computing. As an alternative, various researchers have proposed different ways to exploit the most recent advances in neuroscience about synaptic plasticity [1], especially STDP [16, 11], usually presented as the Hebb rule, revisited in the context of temporal coding. A current trend is to propose computational justifications for plasticity-based learning rules [6, 25, 5, 19]. But the fact that no specific method does emerge from a rather confusing landscape mainly prevents engineers to adopt SNNs for practical applications. Two noticeable attempts in that direction are worth mentioning: one from Jaeger [10], with the Echo State Network (ESN), and the other one from Maass, Natschläger and Markram [14], with the Liquid State Machine (LSM), but in both cases, the performance is hard to control and the models are not yet fully mature.

Here we propose a SNN built on biological bases, with synaptic plasticity, but conceived for supervised classification, a classical machine learning purpose. The network architecture is a set of neurons, without preimposed topology and with sparse connectivity, as in ESN and LSM. The learning rule is a combination of two algorithms for parameter modification: an unsupervised adaptation of weights by synaptic plasticity (STDP) and a supervised adaptation of synaptic transmission delays towards two output neurons, according to a margin criterion. Two ideas motivate the latter rule: First, several complexity analyses of SNNs have proved the interest of programmable delays for computational power [12, 21] and learnability [15]; Second, Izhikevich [9] recently pointed out the activation of polychronous groups, based on the variability of transmission delays inside an STDP-driven set of neurons (see Section 5.2 for details), and proposed that the emergence of several polychronous groups, with persistent activation, could represent a stimulation pattern.

The basic idea of our model is to adapt the output delays in order to enhance the influence of the groups activated by a given pattern towards an output neuron corresponding to the pattern class, and to decrease the influence of the same groups towards the neuron associated to the opposite class. A margin criterion is applied, via a stochastic iterative learning process, for strengthening the separation between the spike-timing of two output neurons coding for two opposite classes.

Section 2 describes the model of SNN and Section 3 defines the learning mechanism. The performance of the model for a classification task is studied through experiments related in Section 4. Section 5 explains the notion of polychronization and its links with the notion of cell assemblies. Section 6 explains the internal behavior of the model and proposes a discussion which could provide a new insight on perceptive memory representation inside the brain.

2 Spiking Classifier

The classifier is a set of M cells (internal network), interfaced with a layer of K input cells and two output cells, one for each class (Figure 1). The network is fed by input vectors of real numbers, represented by spikes in temporal coding: the higher the value, the earlier the spike emission towards the network. For clarity in experiments, successive inputs are presented in large temporal windows, without overlapping input spike emissions from a pattern to the next. The number of the first firing output cell provides the class number, as an answer of the network to the input pattern.

Each cell (input, output, internal network) is a spiking neuron (Section 2.1). Each synaptic connection, from neuron N_i to neuron N_j, is defined by a weight w_{ij} and a transmission delay d_{ij}. The internal network

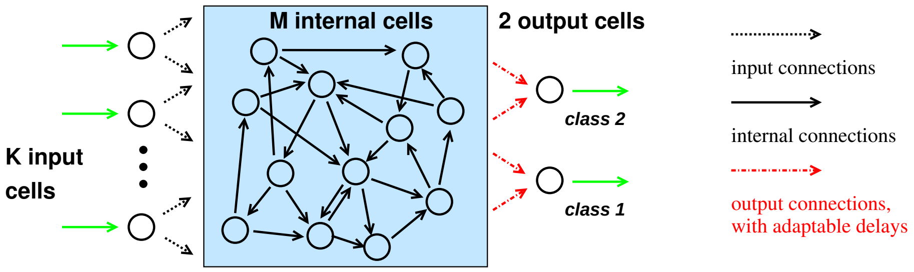


Figure 1: Architecture of the classifier network and interface (green links) with environment.

is composed of 80% excitatory neurons and 20% inhibitory neurons. The internal connectivity is random and sparse, with probability 0.3 for a connection to link N_i to N_j, for all $(i, j) \in \{1, ..., M\}^2$. For pattern stimulation, the input cells are connected to the internal cells with probability 0.1. For class detection, the internal cells are fully connected to each output neuron.

2.1 Model of Neuron

The neuron model is an SRM_0 ("Spike Response Model"), as defined by Gerstner [8], where the state of a neuron N_j is dependent on its last spike time $t_j^{(f)}$ only. The next firing time of N_j is governed by its membrane potential $u_j(t)$ and its threshold $\theta_j(t)$. Both variables are functions of the last firing times of the neurons N_i belonging to the set Γ_j of neurons presynaptic to N_j:

$$u_j(t) = \underbrace{\eta(t - t_j^{(f)})}_{threshold\ kernel} + \sum_{i \in \Gamma_j} w_{ij} \underbrace{\epsilon(t - t_i^{(f)} - d_{ij})}_{potential\ kernel} \quad \text{and} \quad u_j(t) \geq \vartheta \Longrightarrow t_j^{(f+1)} = t \tag{1}$$

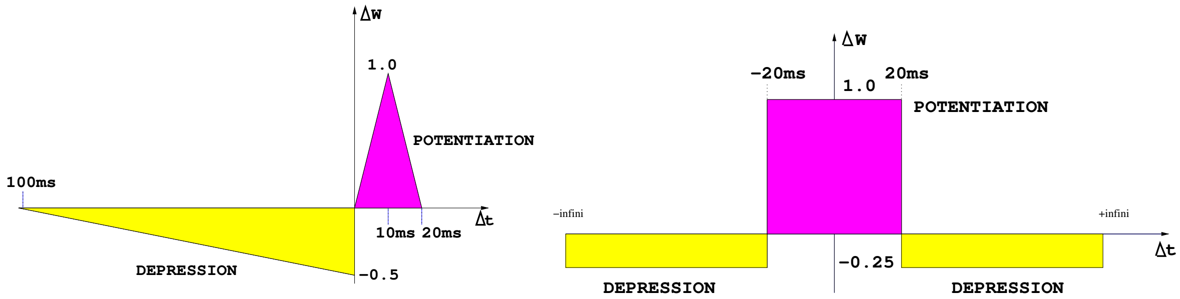
where the potential kernel is modelled by a Dirac increase in 0, followed by an exponential decrease, from value $u_{max} = 8mV$ in 0^+ towards 0, with a time constant $\tau_m = 2ms$. The firing threshold ϑ is set to $-50mV$ and the threshold kernel simulates an absolute refractory period $\tau_{abs} = 7ms$, when the neuron cannot fire again, followed by a reset to the resting potential $u_{rest} = -65mV$ (relative refractory period is not simulated). The simulation is computed in discrete time, with $1ms$ steps. The variables of neuron N_j are updated at each new impact of incoming spike, which is sufficient for computational purpose.

2.2 Synaptic Plasticity

The weight w_{ij} of a synapse from neuron N_i to neuron N_j can be adapted by STDP (Spike-Time-Dependent Plasticity), a form of synaptic plasticity based on the respective order of pre- and postsynaptic firing times. For excitatory synapses, if a causal order (pre- just before post-) is respected, then the strength of the connection is increased. Conversely the weight is decreased if the presynaptic spike arrives at neuron N_j just after a postsynaptic firing, and has probably no effect, due to the refractory period of N_j. Temporal windows, inspired from neurophysiological experiments by Bi & Poo [3], help to calculate the weight modification ΔW as a function of the time difference $\Delta t = t_{post} - t_{pre} = t_j^{(f)} - (t_i^{(f)} + d_{ij})$ as can be computed at the level of neuron N_j.

Following [20], in order to avoid a saturation of the weights to the extremal values $w_{min} = 0$ and $w_{max} = 1$, we apply a multiplicative learning rule in order to take ΔW into account in the weight update rule: See formula in figure 2, where α is a positive learning rate.

For excitatory synapses as well as for inhibitory synapses, a similar principle is applied, and only the temporal window differs (see figure 2). In experiments, we fix $\alpha = \alpha_{exc} = \alpha_{inh} = 0.1$.



if $\Delta t \leq 0$ then decrease the weight:
$w_{ij} \leftarrow w_{ij} + \alpha * (w_{ij} - w_{min}) * \Delta W$

if $\Delta t \geq 0$ then increase the weight:
$w_{ij} \leftarrow w_{ij} + \alpha * (w_{max} - w_{ij}) * \Delta W$

Figure 2: Asymmetrical STDP temporal window (from [17]) for excitatory (left) and inhibitory (right) synapse adaptation.

3 Learning Mechanisms

There are two concurrent learning mechanisms in the model: an unsupervised learning of weights by STDP, operating in the millisecond range, at each new impact t_{pre} or emission t_{post} of a spike, and a supervised learning of output delays, operating in the range of $100ms$, at each pattern presentation.

3.1 STDP Implementation

STDP is applied to the weights of internal cells only. The weights of connections from input layer to internal network are kept fixed, with value $w_{IN} = 3$. The weights of connections from internal network to output neurons are kept fixed, with value $w_{OUT} = 0.5$.

Neuroscience experiments [24] give evidence to the variability of transmission delay values, from $0.1ms$ to $44ms$. In the present model, the delays d_{ij} take integer values, randomly chosen in $\{1, \ldots, 20\}$, both in the internal network and towards output neurons, whereas the delays from input layer to internal network have a zero value, for an immediate transmission of input information.

A synaptic plasticity rule like STDP could be applied to delay learning, as well as to weight learning, but the biological plausibility of such a plasticity is not yet so clear in neuroscience [22]. Hence we do not apply STDP plasticity to delays, but we switch to machine learning in designing a supervised mechanism, based on a margin criterion, for adapting the output delays of our model.

3.2 Delay Adaptation Algorithm

The refractory period of the output neurons has been set to $\tau_{abs}^{out} = 80ms$, such that each output neuron fires once and only once inside a $100ms$ temporal window dedicated to an input pattern presentation. The goal of the supervised learning mechanism we propose here is thus to modify the delays from active internal neurons to output neurons in such a way that the output neuron corresponding to the target class fires before the one corresponding to the non-target class. Moreover, as it has been shown in the machine learning literature, maximizing a margin between the positive and the negative class yields better expected generalization performance [26]. More formally, we thus try to minimize the following criterion:

$$C = \sum_{p \in \text{class1}} |t_1(p) - t_2(p) + \epsilon|_+ + \sum_{p \in \text{class2}} |t_2(p) - t_1(p) + \epsilon|_+ \quad (2)$$

where $t_i(p)$ represents the firing time of output neuron i answering to input pattern p, ϵ represents the minimum delay margin we want to enforce between the two firing times, and $|z|_+ = \max(0, z)$. Hence, for input patterns belonging to class 2, we want output neuron 2 to fire at least ϵ milliseconds before output neuron 1 (Figure 3).

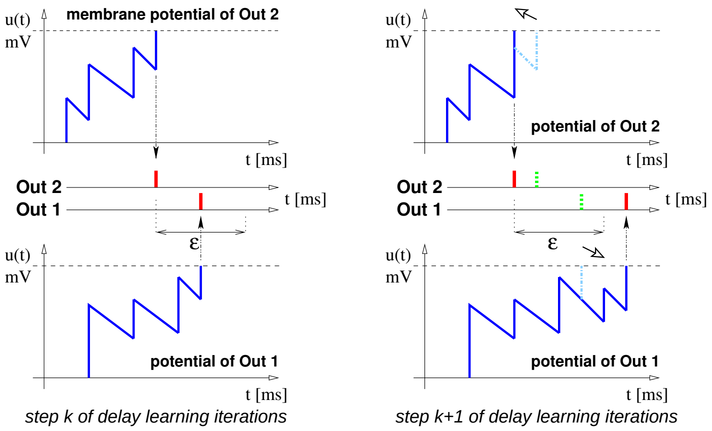


Figure 3: Membrane potential of the two output neurons and effect of one iteration of delay learning on their respective firing times, under the hypothesis that the desired class is 2. Note that, without loss of generality, the curves of exponential decrease have been simplified into straight oblique lines, to represent variations of $u(t)$.

In order to minimize this criterion, we adopt a stochastic training approach, iterating the loop:

```
 after the presentation of a given input pattern p,
   if the difference of firing times between the target and the non-target output
neurons is higher than ε,
     then the pattern is well classified and we do nothing,
     otherwise, for each output neuron, we select the connection that received the
decisive impact, among the last presynaptic neurons responsible for the output spike,
and we decrement its delay if it corresponds to a connection going to the target class
and increment it otherwise (see Figure 3).
```

Hence, at each step, we decrease the probability of an error in the next answer to a similar input.

4 Classifier Performance

A spike raster plot presents all the firing times of all neurons: neuron index with respect to time (in ms) with $K = 10$ input neurons (bottom), followed by $M = 100$ internal neurons (including 20 inhibitory neurons), for the experiments presented here. Firing times of the two output neurons are isolated at the top. A run starts with initial weights $w_{ij} = 0.5$ for all connections in the internal network. Random patterns are presented in input all along the first $300ms$, thus generating a high disordered activity in the network (figure 4). Output neurons spike simultaneously, as soon as their refractory period ends. Due to STDP, the internal activity slowly decreases until complete silence around $1750ms$.

Afterwards, a learning phase is run, between T_{L1} and T_{L2}, with successive alternated presentations of two input patterns (similar to Izhikevich stimulation patterns, Figure 12 in [9]), that represent examples for class 1

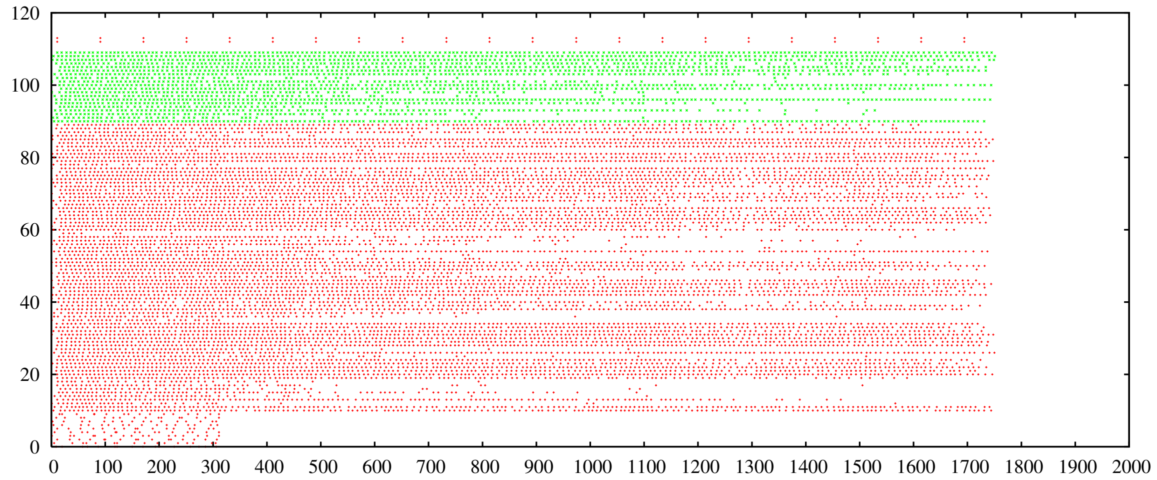


Figure 4: Spike raster plots, for initial random stimulation, from time 0 to 2000.

and class 2 respectively (see Figure 5). Finally, between T_{G1} and T_{G2}, a generalization phase is run, with noisy patterns: Each spike time occurs at $t \pm \eta$ where t is the firing time of the corresponding input neuron for the example pattern of the same class and η is some uniform noise.

4.1 Learning

As can be observed on Figure 5, the internal network activity quickly decreases and then stabilizes on a persistent alternative between two different spike-timing patterns (lasting slightly longer than the time range of pattern presentation), one for each class.

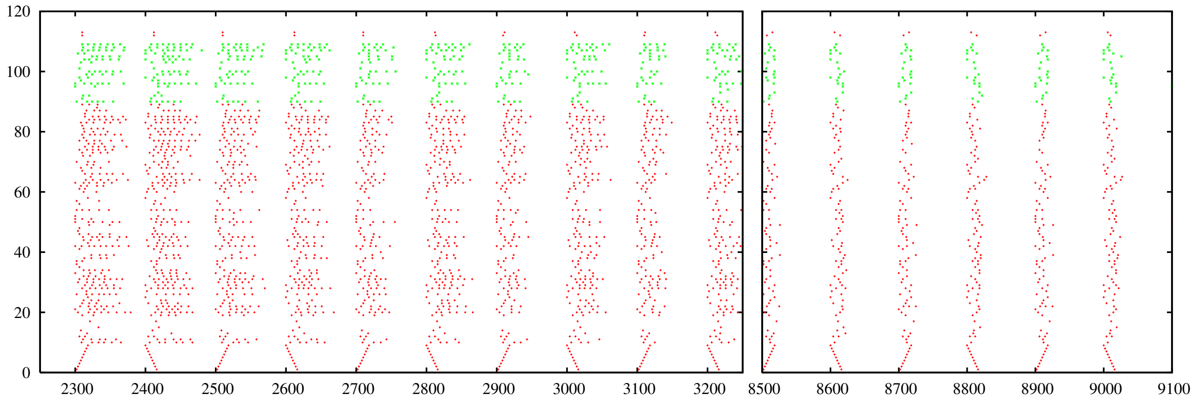


Figure 5: Spike raster plots, for two time slices of a learning run, one just after T_{L1} (left) and the other a long time after activity has began to stabilize (right).

The evolution of the firing times of the two output neurons reflects the application of the delay adaptation algorithm. Starting from simultaneous firing, they slightly dissociate their responses, from a pattern presentation to the next, according to the class corresponding to the input (top of left frame, Figure 5). In the right frame of the figure, the time interval separating the two output spikes has become larger, due to delay adaptation, and is stable, since the margin ϵ has been reached. The internal activity is quite invariant, except seldom differences due to the still running STDP adaptation of weights (this point will be further discussed in Section 6).

4.2 Generalization

On Figure 6, two noise levels can be compared. Although the internal network activity is clearly disrupted, the classification performance remains good: average success rate, on 100 noisy patterns of each class, is 96% for $\eta = 4$, when noisy patterns are presented alternatively, class 2 after class 1, and still 81% for $\eta = 8$, where the input patterns are hard to discriminate by a human observer. We observed a slight effect of sequence learning: only 91% and 73% success respectively, when class 1 and class 2 are presented in random order (this point remains to be investigated in future work).

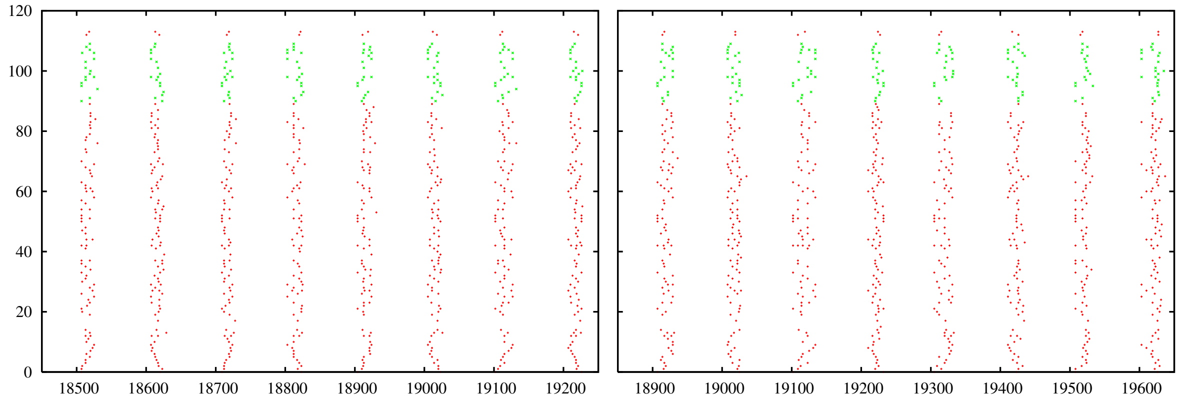


Figure 6: Spike raster plots, for two series of noisy patterns: $\eta = 4$ (left) and 8 (right).

We observe that the obtained margin between the two output firing times can be less or more than ϵ. For each pattern, this margin could be exploited as a confidence measure on the network answer. Moreover, most of the non-successful cases are due to simultaneous firing of the two output neurons (on Figure 6, only one wrong order near the left of $18800ms$). Such ambiguous responses can be considered as "non-answers", and could lead to define a subset of rejected patterns. Wrong order output spike-firing patterns are seldom, which attest the robustness of the learning algorithm.

4.3 Weight Distributions

In order to illustrate the weight adaptation that occurs in the internal network, Figures 7 and 8 show respectively the distribution of excitatory and inhibitory weights quantized into 10 uniform segments of 0.1 and captured at different instants. The distribution at time 0 is not shown, as all weights were initialized to 0.5. Excitatory weights (Figure 7) tend to be Gaussian around the original distribution (time 300 and 2000). We have measured that the average amount of time between two spikes during the first $1700ms$ corresponds to $8ms$. In the excitatory STDP temporal window (Figure 2, top left), $|\Delta W|$ in the range of $8ms$ is comparable at both sides of 0, and thus explains this Gaussian redistribution. Then weights are mainly uniformly distributed from 0 to 0.7 at time 4000. They finally equilibrate (time 10000) with approximately 50% of weights very close to 0, other weights being decreasingly distributed from 0.1 to 0.7.

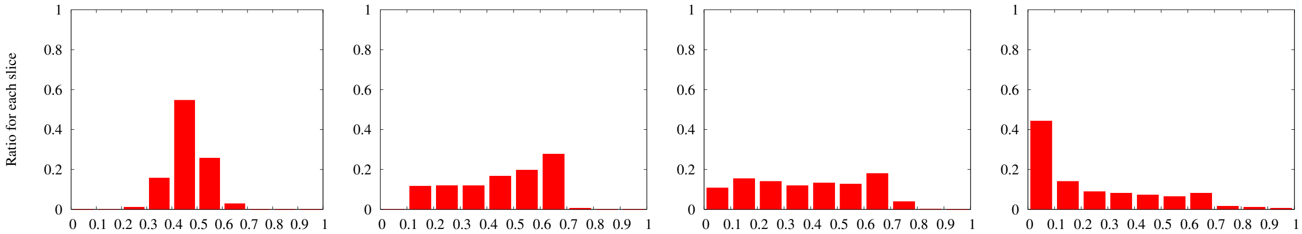


Figure 7: Excitatory weights distribution at time 300, 2000, 4000, 10000

Let us now consider inhibitory weights in Figure 8. As the initial internal activity is strong, the weights are modified in a very short time range. Indeed, looking at time 300 (Figure 8, left) we see that weights have already nearly all migrated to a very high value (close to 1). This surprising violent migration can as well be explained by the inhibitory STDP function, where close spikes in an inhibitory synapse produce a weight potentiation (see Figure 2, top right). After the initial stimulation stopped, weights begin to redistribute as the internal network activity slows down. From then on, weight distribution has reached a state that will continue to evolve, until time 10000, when distribution becomes very similar to time 17000 (end of learning phase).

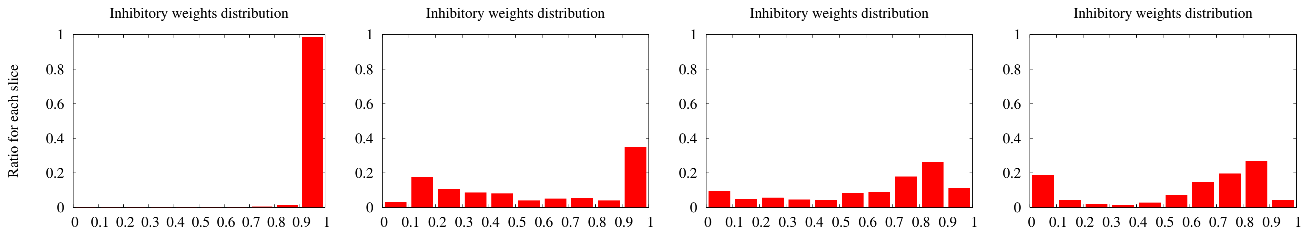


Figure 8: Inhibitory weights distribution at time 300, 2000, 4000, 10000

5 Polychronization

Common thought that interactions between neurons are governed by their mean firing rates has been the basis of most traditional artificial neural network models. Since the end of the 90's, there is a growing evidence, both in neuroscience and computer science, that precise timing of spike emission is a central feature in cognitive processing. However, although reproducible spike-timing patterns have been observed in many physiological experiments, the way these spike-timing patterns, at the millisecond scale, are related to high-level cognitive processes is still an open question.

Deep attention has been paid to synchronization of firing times for subsets of neurons inside a network. The notion of synfire chain [2, 7], a pool of neurons firing synchronously can be described as follows: if several neurons have a common postsynaptic neuron N_j and if they fire synchronously then their firing will superimpose in order to trigger N_j. However, the argument falls down if the axonal transmission delays are to be considered, since the incoming synapses of N_j have no reason to share a common delay value.

5.1 Cell Assemblies

A cell assembly can be defined as a group of neurons with strong mutual excitatory connections. Since a cell assembly tends to be activated as a whole once a subset of its cells are stimulated, it can be considered as an operational unit in the brain. An association can be viewed as the activation of an assembly by a stimulus or another assembly. In this context, short term memory would be a persistent activity maintained by reverberations in assemblies, whereas long term memory would correspond to the formation of new assemblies, e.g. by a Hebb's rule mechanism. Inherited from Hebb, current thoughts about cell assemblies are that they could play a role of "grandmother neural groups" as basis of memory encoding, instead of the old debated notion of "grandmother cell", and that material entities (e.g. a book, a cup, a dog) and, even more, ideas (mental entities) could be represented by different cell assemblies.

5.2 Polychronous Groups

Synchronization appears to be a too restrictive notion when it comes to grasp the full power of cell assemblies processing. This point has been highlighted by Izhikevich [9] who proposes the notion of polychronization. Based on the connectivity between neurons, a polychronous group is a possible stereotypical time-locked firing pattern. For example, in Figure 9, if we consider a delay of $15ms$ from neuron N_1 to neuron N_2, and a delay of $8ms$ from neuron N_3 to neuron N_2, then neuron N_1 emitting a spike at time t and neuron N_3 emitting at time $t+7$ will trigger a spike emission by neuron N_2 (supposing two coincident spike impacts enough to make a neuron spike). Since neurons of a polychronous group have matching axonal conduction delays, the group can be the basis of a reproducible spike-timing pattern: firing of the first few neurons with the right timing is enough to activate most of the group.

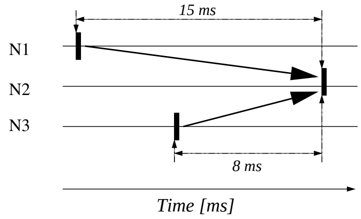


Figure 9: Example of two trigger neurons

Since any neuron can be activated within several polychronous groups, at different times (e.g. neuron 76 in Figure 10), the number of coexisting polychronous groups in a network can be much greater than its number of neurons, thus opening possibility of high memory capacity. We have detected 104 potentially activable polychronous groups in a network of $M = 100$ neurons, and more than 3000 in a network of $M = 200$ neurons.

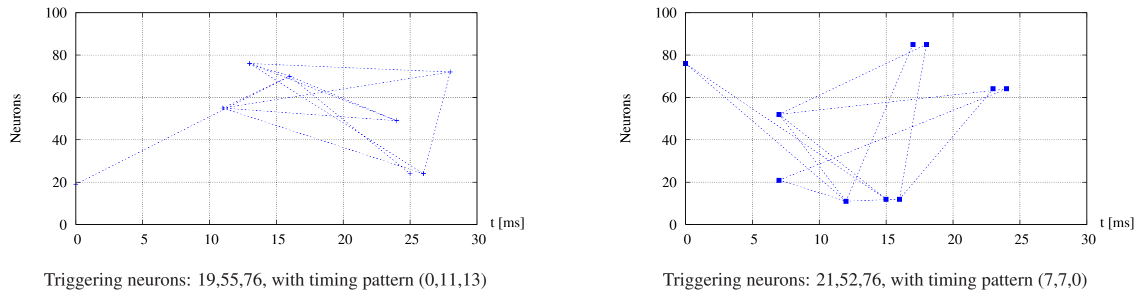


Figure 10: Two polychronous groups among 104 detected groups: Starting from the three initial triggering neurons, further neurons of the group can be activated, in chain, with respect to the spike-timing patterns represented on the diagrams.

Our model proposes a way to confirm the link between an input presentation and persistent spike-timing patterns inside the network, and the way we take advantage of polychronous groups for deriving our supervised learning algorithm for classification is explained in the next section.

6 Network Internal Behavior

Figure 11 presents the evolution of polychronous groups actually activated in our experiments. All the potential polychronous groups in the internal network, depending on its topology and the values of the internal transmission delays (that are kept fixed), have been inventoried (ordinate numbers, on Figure 11). We have referenced only the polychronous groups with 3 triggering neurons (see Figure 10 for examples). The evolution of activated polychronous groups is entirely governed by STDP, the only learning process acting inside the internal network.

6.1 Learning Process Justification

We observe that many polychronous groups are frequently activated during the initial random stimulation that generates a strong disordered activity in the internal network (before $2000ms$). At the beginning of the learning phase (which goes from $2000ms$ to $17000ms$), many groups are activated, and then, roughly after $5000ms$, the activation landscape becomes very stable, with a few specific polychronous groups associated to each class: groups 3, 41, 74, 75 and 83, switching to 85, for class 1, whereas groups 12, 36, 95, sometimes 99, and 49, switching to 67, for class 2.

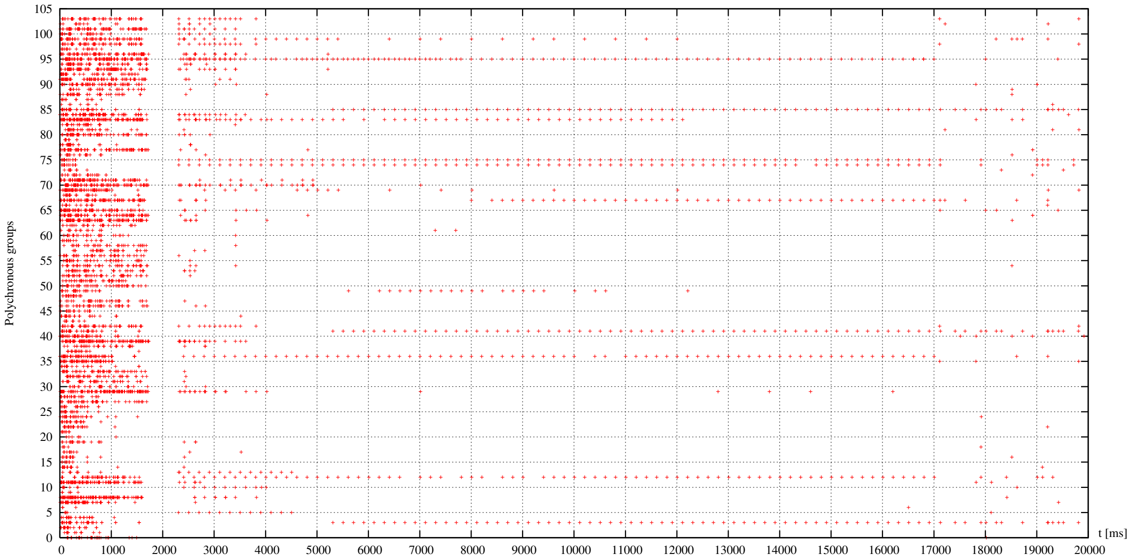


Figure 11: Evolution of polychronous groups that are activated during the learning and generalization phases of experiments reported in Section 4.

Several interesting observations can be reported. As noticed by Izhikevich, there exist groups that start to be activated only after a large number of repeated stimulations (e.g. 41, 49, 67 and 85), but we also observe that other groups stop their activation after a while (e.g. 5 and some others [activated until $4000/5000ms$ only], 49, 70, 83 and 99 [later]). A very interesting case is the polychronous group number 95 which is first activated by both example patterns, and then (around time $7500ms$) stops responding for class 1, thus specializing its activity for class 2. Such phenomenon validates that synaptic plasticity provides the network with valuable adaptability.

The influence of activated polychronous groups on the learning process can be clearly exhibited. We have registered (Figure 12) the indices of the presynaptic neurons responsible for the application of the output delay update rule, at each iteration where the example pattern is not yet well classified (cf. algorithm, Section 3). For instance, neuron $\#42$, which is repeatedly responsible for delay adaptation, is one of the triggering neurons of the polychronous group number 12, activated for class 2 during the training phase. One will notice that delay adaptation stops before learning phase is over, which means learning process is already efficient around time 10000

In the generalization phase (after $17000ms$), the main and most frequently activated groups are those identified during the learning phase. This observation accredits the hypothesis that polychronous groups are representative of the class encoding realized by our two-scale learning mechanism.

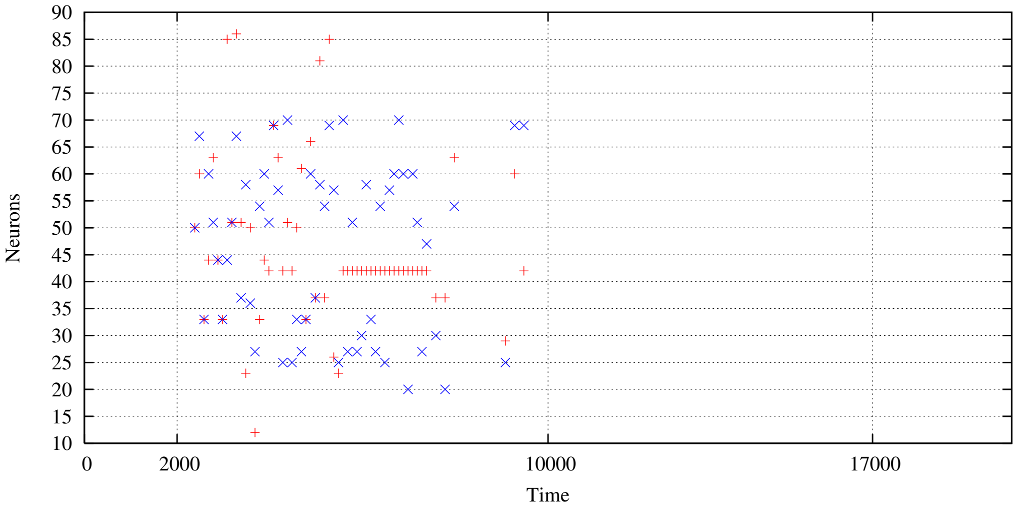


Figure 12: Neurons that triggered a delay change. Figure focuses on internal excitatory neurons: inhibitory neurons wouldn't produce a delay shift as they wouldn't trigger a spike of an output neuron. Red "+": class 1; blue "x": class 2

6.2 Polychronous Groups vs Cell Assemblies

The behavior of our internal network could bring new arguments in the debate about the definition of cell assemblies and their role in percept memory processing. We observe that the subsets of inactive / highly active neurons are nearly the same for the two different input patterns (see Figure 5, left) and that the phenomenon remains, even when learning is achieved (right). Hence, a spatial definition of cell assemblies fails to validate the hypothesis of different assemblies coding for different input patterns. However, the spike-timing patterns of polychronous groups appear to be clearly characteristic of the pattern class encoding. These observations could give new tracks for neurophysiological investigations, in order to understand the underlying processes that govern percept memorization.

7 Conclusion

With supervised classification as a goal task, we have proposed a two-scale learning mechanism for spiking neuron networks, with unconstraint topology. This mechanism presents several interests. First, the algorithm for delay adaptation is computationally easy to implement. Second, the way the learning algorithm operates can be well explained by the concept of polychronization and the internal network is no longer a black-box, contrary to the ESN or LSM models. Third, the model helps to better understand the functional links between the percept memorization process and the activation of internal neurons, thus enhancing the advantage of the concept of spike-timing patterns over the idea of spatial cell assemblies. Finally, the model has shown promising performance for learning and generalization on a classification task. This latter point is currently being further tested in larger scale experiments using realistic benchmark input patterns. Hence for instance, initial experiments on a two-class version of the well known USPS digit dataset yielded encouraging results.

Acknowledgement

The authors acknowledge David Meunier for precious help and clever advice about the most pertinent way to implement STDP.

References

[1] L.F. Abbott and S.B. Nelson. Synaptic plasticity: taming the beast. *Nature Neuroscience*, 3:1178–1183, 2000.

[2] M. Abeles. *Corticonics: Neural Circuits of the Cerebral Cortex*. Cambridge Univ. Press, 1991.

[3] G.-q. Bi and M.-m. Poo. Synaptic modification in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and polysynaptic cell type. *J. of Neuroscience*, 18(24):10464–10472, 1998.

[4] S.M. Bohte, Kok J.N., and H. La Poutré. Spikeprop: Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48:17–37, 2002.

[5] S.M. Bohte and M.C. Mozer. Reducing spike train variability: A computational theory of Spike-Timing Dependent Plasticity. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *NIPS*2004, Advances in Neural Information Processing Systems*, volume 17, pages 201–208. MIT Press, 2005.

[6] G. Chechik. Spike-timing dependent plasticity and relevant mutual information maximization. *Neural Computation*, 15(7):1481–1510, 2003.

[7] M. Diesmann, M.-O. Gewaltig, and A. Aertsen. Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402:529–533, 1999.

[8] W. Gerstner and W. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

[9] E.M. Izhikevich. Polychronization: Computation with spikes. *Neural Computation*, 18(1):245–282, 2006.

[10] H. Jaeger. The “echo state” approach to analysing and training recurrent neural networks. Technical Report TR-GMD-148, German National Research Center for Information Technology, 2001.

[11] R. Kempter, W. Gerstner, and J. L. van Hemmen. Hebbian learning and spiking neurons. *Physical Review E*, 59(4):4498–4514, 1999.

[12] W. Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10:1659–1671, 1997.

[13] W. Maass and T. Natschläger. Networks of spiking neurons can emulate arbitrary Hopfield nets in temporal coding. *Network: Computation in Neural Systems*, 8(4):355–372, 1997.

[14] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[15] W. Maass and M. Schmitt. On the complexity of learning for a spiking neuron. In *COLT'97, Conf. on Computational Learning Theory*, pages 54–61. ACM Press, 1997.

[16] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275:213–215, 1997.

[17] D. Meunier and H. Paugam-Moisy. Evolutionary supervision of a dynamical neural network allows learning with on-going weights. In *IJCNN'2005, Int. Joint Conf. on Neural Networks*, pages 1493–1498. IEEE–INNS, 2005.

[18] T. Natschläger and B. Ruf. Spatial and temporal pattern analysis via spiking neurons. *Network: Comp. Neural Systems*, 9(3):319–332, 1998.

[19] J.-P. Pfister, T. Toyoizumi, D. Barber, and W. Gerstner. Optimal spike-timing dependent plasticity for precise action potential firing in supervised learning. *Neural Computation*, 18(6):1318–1348, 2006.

[20] J. Rubin, D.D. Lee, and H. Sompolinsky. Equilibrium properties of temporally assymmetric Hebbian plasticity. *Physical Review Letters*, 89(2):364–367, 2001.

[21] M. Schmitt. On computing boolean functions by a spiking neuron. *Annals of Mathematics and Artificial Intelligence*, 24:181–191, 1998.

[22] W. Senn, M. Schneider, and B. Ruf. Activity-dependent development of axonal and dendritic delays, or, why synaptic transmission should be unreliable. *Neural Computation*, 14:583–619, 2002.

[23] J. Sima and J. Sgall. On the nonlearnability of a single spiking neuron. *Neural Computation*, 17(12):2635–2647, 2005.

[24] H.A. Swadlow. Physiological properties of individual cerebral axons studied in vivo for as long as one year. *Journal of Neurophysiology*, 54:1346–1362, 1985.

[25] T. Toyoizumi, J.-P. Pfister, K. Aihara, and W. Gerstner. Generalized Bienenstock-Cooper-Munro rule for spiking neurons that maximizes information transmission. *Proc. Natl. Acad. Sci.*, 102(14):5239–5244, 2005.

[26] V.N. Vapnik. *Statistical learning theory*. Wiley, 1998.