

# A Learning Algorithm for Synfire Chains

Jacques Sougné

## Abstract

Neurobiological studies indicate very precise temporal behavior of neuron firings. Abeles [1] has recorded spike timing of different cortical cells and, in particular, has observed the following level of precision: when a neuron A fires, neuron B would fire 151ms later and neuron C would fire precisely 289ms after that—with a precision across trials of 1 ms! Such long delays require dozens of combined transmission delays from the presynaptic neuron (A) to the postsynaptic neuron (C). The mechanism proposed by Abeles for generating such precise delayed synchronization has been called *synfire chains*. How could synfire chains develop? What learning procedure could generate such precise temporal chains? How could a connectionist network of spiking neurons learn synfire chains? An algorithm for a network of spiking neurons that learns synfire chains will be presented.

## 1. Introduction

The human brain is extremely sensitive to time. According to Abeles et al. [2], people hearing a sound from a source situated 1 degree to the right of the centerline of their face, at a distance of 1 meter, will receive the acoustic signal at their right ear 12  $\mu$ s before the left ear. This means that detecting the source of a sound requires very accurate temporal detection. According to Calvin [4], throwing a projectile at a 20cm wide target, from a distance of 7 meters, requires a release time accuracy of less than 1 ms. Findings on synfire chains [1, 2, 17] indicate very precise temporal behavior of neuron firings. Researchers have recorded spike timing of different cortical cells in monkeys. In particular, they observed the following stimulus-dependent kind of pattern: when a neuron *a* fired, neuron *b* would fire 151ms later, and neuron *c* would fire 289ms later with a precision across trials of 1 ms! Delays of this duration would require dozens of transmission delays from *a* to *c* neuron. There are two hypotheses about how this phenomenon could occur. The first is based on an increase in a population rate which build excitation in another population which, in turn, increases its firing rate, etc. The second is the mechanism proposed by Abeles [1] which has been called *synfire chains*. Since cortical synapses are relatively weak, many inputs to cells must arrive at the same time for them to fire. Consequently, each step in the synfire chain requires a pool of neurons whose firings simultaneously raise the potential of the next pool of neurons to allow them to fire. In this mechanism each cell of the chain fire only once. Recent experiments [17] indicate that these precise firing sequences correlate more to behavior than rate modulation and does not seems to be a byproduct of rate modulation. These results seems to confirm the synfire chain hypothesis.

How could synfire chains develop? What learning procedure could generate such precise temporal chains? How can a connectionist network of spiking nodes learn and display synfire chains? This is the fundamental part of the present study.

A network of spiking nodes (INFERNET) is provided with a learning algorithm that allows it to learn synfire chains.

## 2. Learning Synfire Chains

Previous work on learning synfire chains has focused on how they can develop from a chaotic net with an unsupervised Hebbian learning rule [3, 10, 11]. These studies involved an external stimulus which makes a pool of neurons fire at time 0. Subsequently a sequence of successive pools of neuron firings occurs engendered by the random connection weights of the network. Active connections are modified by a Hebbian learning rule. After learning, when the same stimulus is presented, the same chain of neuron firing is observed, thereby constituting a synfire chain. These studies show that these chains are stable, noise tolerant and that one network can store many different chains. Formal analysis showed that there is a relation between the network size and the length of learnable synfire chains [3, 10], and that the recall speed should be faster than the training speed [21].

This paper explores how a synfire chain can develop to link two pools of neuron firings provoked by two sequential external stimuli. After learning, the first external stimulus should activate the stored synfire chain.

## 3. INFERNET

INFERNET [18, 19, 20] is a network of spiking neurons [13]. In INFERNET, nodes can be in two different states: they can fire (on), or they can be at rest (off). A node fires at a precise moment and transmits activation to other connected nodes with some time course. When a node activation or potential  $V_i^{(t)}$  reaches a threshold, it emits a spike. After firing, the potential is reset to some resting value  $V_r$ . Inputs increase the node potential, but some part of the node potential is lost at each time step. Spiking neuron models use a post synaptic potential (PSP) function.

INFERNET is not a fully connected network; its structure is organized by clusters of nodes which constitute subnets. Each subnet is fully connected. From each node of a subnet there is a connection to every other node within that subnet. Some subnet nodes have connections to external subnet nodes. It not only reduces the computational demands of the program, but also better corresponds to the actual organization of the brain. The brain is not fully connected; in particular, there are more intra-cortical connections than inter-cortical connections. Each connection is either excitatory or inhibitory. Excitatory connections increase the potential of postsynaptic nodes, while inhibitory connections decrease their potential. Two variables affect each connection: weight and delay. Each weight corresponds to the synaptic strength between a presynaptic and postsynaptic cell. The weight between a presynaptic node  $j$  and a postsynaptic node  $i$  is designated by  $w_{ij}$ . Noise is added to this value and the resulting noisy connection is denoted by  $\hat{w}_{ij}$ . The delay  $d$  of a connection determines when the effect of the presynaptic node firing will be maximum on the postsynaptic node. There is also a noise factor on the delay. The noisy delay is denoted by  $\hat{d}$ . This delay corresponds to the axonal, synaptic and dendritic delays of real neurons.

A signal, whether excitatory or inhibitory, will be affected by a leakage factor. When the signal has reached its maximum, at each following step of 1 ms, the signal will be divided by 2. Delays and leakage factors define the Excitatory Post Synaptic Potential (EPSP) or Inhibitory Post Synaptic Potential (IPSP) functions shown in Figure 1a. The y-axis refers to the postsynaptic node potential  $V_i$ . The x-axis is the time difference in ms between the time  $t$  and the time of the presynaptic node firing  $t_j^{(f)}$ . The resulting postsynaptic (PSP) equation  $\varepsilon_{ij}(x)$  is given by:

$$\varepsilon_{ij}(x) = \frac{1}{2^x} H(x) \quad (1)$$

$$\text{where: } H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (2)$$

and  $x$  is the difference between the time  $t$ , the time of the presynaptic node firing, and the noisy delay on the connection:  $x = t - t_j^{(f)} - d$ .

When a node potential  $V_i$  reaches a threshold  $\theta$ , it emits a spike. Thereafter, the potential is reset to its resting value. Figure 1b illustrates a spike of an INFERNET node. After emitting a spike, a node enters a refractory period. This corresponds to the membrane resistance of real neurons which increases after a spike. In INFERNET, the refractory state of node  $i$  depends only on the last spike of the node  $i$ :  $t_j^{(f)}$ . A value dependent on the refractory state is subtracted from the node state value  $V_i$ . This value is denoted by  $\eta_i(u)$ , where  $u$  is the difference between the current time  $t$  and the time of the last spike of node  $i$ :  $u = t - t_j^{(f)}$ . The shape of this function is shown in Figure 1c.

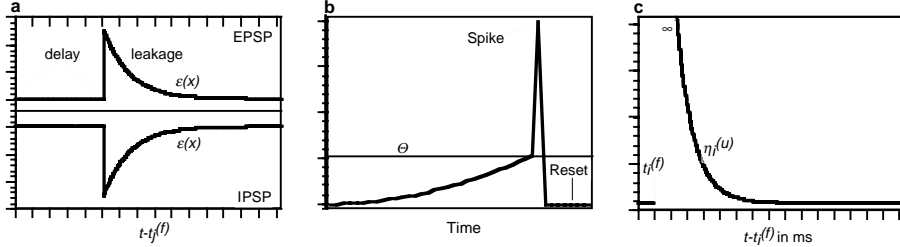


Figure 1, a: EPSP and IPSP function in INFERNET  $\varepsilon(x)$ . b: A spike in INFERNET. c: The function  $\eta_i(u)$  taking into account the refractory period of a node..

All variables affecting the potential of a node have now been defined. Equation (3) express how  $V_i^{(t)}$ , the potential or state of node  $i$ , is calculated at each time step.

$$V_i^{(t)} = \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in F_j} \hat{w}_{ij} \varepsilon_{ij}(x) - \eta_i(u) \quad (3)$$

Node  $i$  fires when its potential  $V_i(t)$  reaches the threshold  $\Theta$ . This potential is affected by connection weights  $\hat{w}_{ij}$  coming from each presynaptic node  $j$ . The set of presynaptic connections to node  $i$  is given by  $\Gamma_i = \{j | j \text{ is presynaptic to } i\}$ .  $F_j$  is the set of all firing times of presynaptic nodes  $j$ :  $t_j^{(f)}$ . Noisy connection weights linking  $j$  node to  $i$  node are  $\hat{w}_{ij}$ . The equations  $\varepsilon_{ij}(x)$  expresses the postsynaptic potential function as shown in Figure 1a. A comment is necessary about the threshold  $\Theta$ . Random noise is added to its value. Assuming a node is no longer in its refractory

period, its firing requires simultaneous full excitatory input from more than one node. In agreement with real neural networks, a single synapse cannot provoke a postsynaptic action potential. A single cortical synapse raises the potential of a postsynaptic neuron by 0.1 to 1 mV [2].

### 3.1 Hebbian Learning

Long term potentiation (LTP) and depression (LTD) are the basic mechanisms of long-lasting modifications of synaptic efficiency. Hebb [9] postulated that when presynaptic activity coincides with postsynaptic activity, the connection between both neurons is strengthened. According to recent experiments, the modification of synaptic efficiency depends on precise timing of afferent signals (neurotransmitters binding to receptors) and the postsynaptic neuron spike. LTP seems to require that postsynaptic action potential be simultaneous or subsequent to EPSP [14, 22]. In short, when the signal from the presynaptic neuron firing arrives before, or during the spike of postsynaptic neuron, the synapse is strengthened (LTP). In contrast, when the signal issued from the presynaptic neuron firing arrives after the spike of postsynaptic neuron, the synapse is depressed (LTD).

### 3.2 Learning in INFERNET

The plasticity of a synapse  $w_{ij}$  is a function of three parameters: the firing time of the presynaptic neuron:  $t_j^{(l)}$ , the transmission delay between this firing and its effect on the postsynaptic neuron ( $d_{ij}$ ), and the firing time of postsynaptic neuron  $t_i^{(l)}$ . Learning in INFERNET consists of modifying the weights of connections between nodes  $w_{ij}$  by a value  $\Delta w_{ij}$ . The INFERNET Hebbian learning function is shown in Figure 2. This function follows empirical studies [14, 22]. Similar functions were used in various simulation experiments [12, 16].

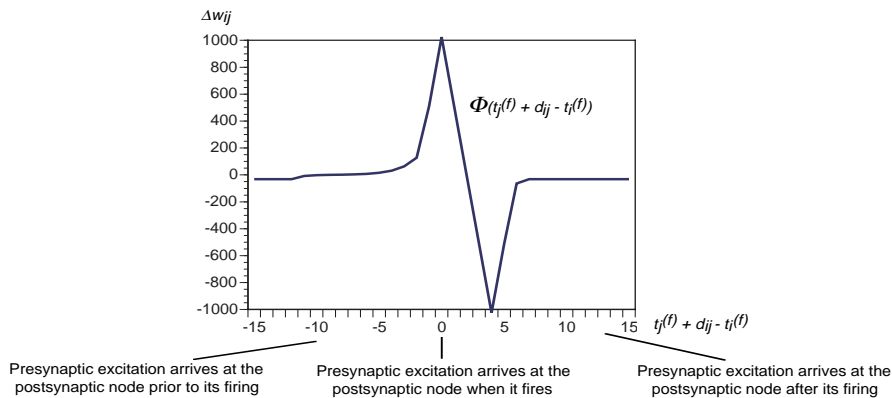


Figure 2: INFERNET Hebbian learning function: when the signal from the presynaptic neuron firing arrives before or during the spike of the postsynaptic neuron, the synapse is strengthened (LTP); when the signal arrives after the spike of the postsynaptic neuron, the synapse is depressed (LTD)

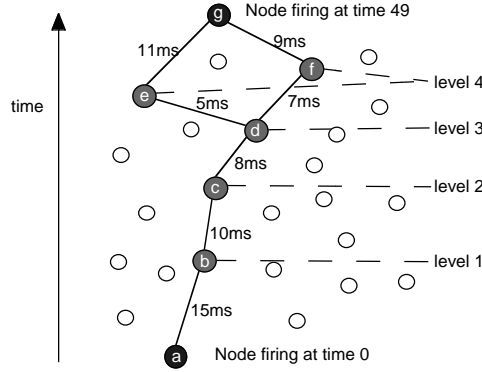


Figure 3: The chaining rule problem: Learning consists of finding a path that makes node  $g$  fire 49 ms later than nodes  $a$  and  $a'$ .

The learning algorithm's job is to reproduce the temporal relation between two successive inputs. This objective is quite difficult because two successive inputs can be separated by several tenths of a second and a single connection cannot alone be responsible for such long delays. A long chain of successive pools of node firings is therefore required. This problem is illustrated in Figure 3. The problem is linking nodes  $a$  and  $a'$  that fire at time 0 with node  $g$  firing at time 49. In the learning phase, only nodes  $a$  and  $a'$  then 49 ms later,  $g$  are externally stimulated. The system has to find a chain of node firing that makes the target node  $g$  fire at time 49 when the probe nodes  $a$  and  $a'$  are firing at time 0. This is the chaining rule problem. The level is defined as the number of steps (or pools of firing nodes) separating the input probe nodes' firing from the target nodes' firing.

### 3.3 The Chaining Rule

The chaining rule is based on the following assumption: We know the delay of signal propagation from a presynaptic node to a postsynaptic node. From the refractory state we know when a particular node fired, we can therefore detect which synapse can make a node fire at the right moment. In figure 3, one can detect which nodes make node  $g$  fire at the right moment ( $e$  and  $f$ ). One can also determine which nodes cause the firing of node  $e$  at the right time ( $d$  and  $d'$ ), and so on. In order to reduce combinatory explosion, only the  $n$  best contributing nodes are selected for the next level in this chaining rule. Connections between nodes will be modified according to equation (4):

$$\Delta w_{ij} = \Phi(t_j^{(f)} + d_{ij} - t_i^{(f)}) - \lambda \quad (4)$$

$$\text{where } \lambda = \begin{cases} -level & \text{if } \Phi(t_j^{(f)} + d_{ij} - t_i^{(f)}) \text{ is negative} \\ level & \text{otherwise} \end{cases} \quad (5)$$

This rule is based on the history of node firing and has some neurobiological justification. For example, the state of a synapse is indicative of its past activity

[15]: a synapse has a memory! Moreover, empirical studies [5] show that LTP also propagates from the originating synapse to neighbouring synapses, giving some plausibility to the chaining rule.

The learning algorithm is triggered only when external input is presented. We can imagine that external input provides a strong signal that triggers the chaining rule. Note that Hebbian learning does not seem to be dependent on this kind of signal and affects synapses after an action potential. Here, the target input is the signal to launch the chaining rule. The objective is to link the probe nodes' firing to the target nodes' firing and to avoid reinforcing other irrelevant firings.

### 3.4 Algorithm

```

For each input node firing  $t_i^{(f)}$ 
  For each presynaptic node  $j$ 
    Calculate  $\Delta w_{ij}$  and add it to  $w_{ij}$ 
    Select the  $n$  best nodes  $\{j'\}$ 
    For each node  $j' \in \{j'\}$ 
      Set level to 1
      For each node  $j''$  presynaptic to node  $j'$ 
        Calculate  $\Delta w_{jj''}$  and add it to  $w_{jj''}$ 
        Select the  $n$  best nodes  $\{j''\}$ 
        For each node  $j'' \in \{j''\}$ 
          Set level to 2
          Etc. up to level 50.

```

## 4. Performance of the Algorithm

In the following experiments two inputs are presented, one (the probe) at time 0 ms and one (the target) some time later. The task for the network is to correctly reproduce the temporal association between these two inputs and therefore build a synfire chain between them. When trained, the network is able to trigger this synfire chain whenever the first input is presented. This ability was taken as a measure of the learning performance. The connections are randomly set and are modified by the learning algorithm, while the delays are randomly set (between 4 and 10 ms for intra subnet connections, and between 10 and 20 ms for inter subnet connections) but do not change. Noise affected thresholds and connection weights but not delays.

### 4.1 Learning Depends on the Lag between the Probe and the Target

In the first simulation, different lags between the probe and the target were tested. Results are shown in figure 4. Different measures were collected: the mean number of epochs (or association presentation) needed for the network to reproduce the association with a 1 ms precision. Figure 4a shows, not surprisingly, that as the lag between the probe and the target increases the number of presentations required increases. Figure 4b shows the proportion of successes on the 10 trials for different lag times. A success is counted if the network stabilize in a way that the target node fire at the right moment with a 1ms precision. Performance drops significantly for the 120ms group. Figure 4c shows the proportion of the successful trials in which

some early and spurious firing occurred and was not removed by the learning algorithm. This emphasizes the difficulty of the task. The algorithm can correctly reinforce a connection (between nodes *a* and *b*) that leads to the probe node's firing at the right time, but could also cause the probe nodes to fire earlier if node *a* fires several times before.

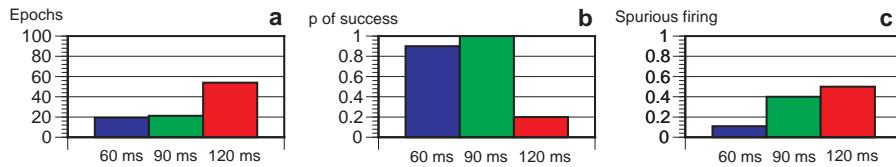


Figure 4: Learning synfire chains of different length. **a**: Number of epochs required for a synfire chain to be learned with a 1 ms precision. **b**: Proportion of learning successes. **c**: Proportion of spurious firing within successful cases.

## 4.2 Learning Depends on the Network Size

In this simulation, different network sizes that had to learn the same synfire chain (a 90 ms chain) were compared. The first network is composed of 800 nodes (50 subnets of 16 nodes), the second network contained 256 nodes (8 subnets of 32 nodes). Ten trials were run for each network. Results shown in figure 5 indicate a strong relation between the net size and the ability to learn synfire chains. The learning algorithm performs better the more nodes there are, providing many alternative paths between the probe nodes and the target nodes. This is consistent with formal analysis [3, 10].

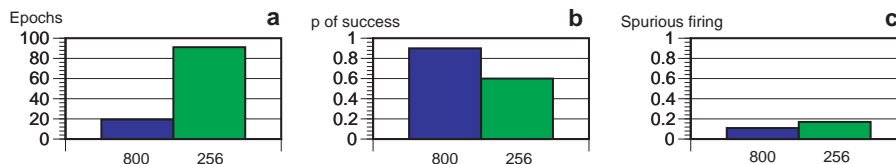


Figure 5: The performance of the algorithm with different network sizes. **a**: Number of epochs required for a synfire chain to be learned with a 1 ms precision. **b**: Proportion of learning successes. **c**: Proportion of spurious firing within successful cases.

## 4.3 The Need for Long Term Depression

This simulation compares the performance of the learning algorithm under three different conditions with ten trials per condition. In the first condition, LTP and LTD were used as described in figure 2, in the second condition LTD was removed but every connection was decayed by a fixed value at every epoch, the last condition only involved LTP. Results shown in figure 6 indicate a need for decaying connection weight. When no decay parameter is used all nodes began to fire too frequently. This is shown by the proportion of spurious firing shown in Figure 6c. Despite the approximately equal performance of the LTD and decay conditions, LTD is to be preferred since decay lowers connection weights blindly and may put the network in a state where it can no longer learn another synfire chain after

learning the first one. This experiment seems to confirm data from Munro & Hernandez [16].

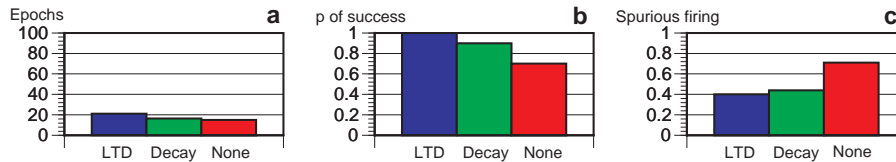


Figure 6: The effect of depression **a**: Number of epochs required for a synfire chain to be learned with a 1 ms precision. **b**: Proportion of learning successes. **c**: Proportion of spurious firing within successful cases.

#### 4.4 Learning Depends on Sparseness of Connections

This simulation explores the effect of density of connection on the algorithm performance. INFERNET is organised in fully connected subnets that have a limited and tunable number of connections to other subnets. Two conditions were compared: the low connection density condition involved 50 subnets with 5 connections between each subnets and the high density condition involved 50 subnets with 6 connections between each subnet. Results shown in figure 7 indicate that the learning algorithm works better when the network is sparsely connected. When connections are too dense, target nodes fire too early and only rarely settle on the desired timing. This is probably due to the fact that there are too many direct paths between the probe nodes and the target nodes.

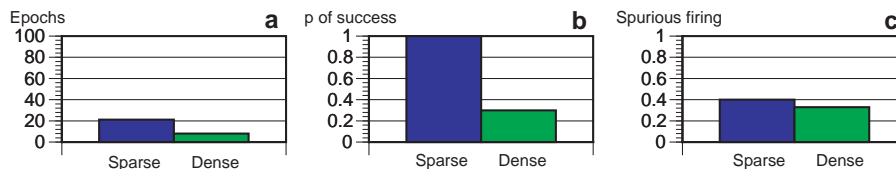


Figure 7: The effect of sparseness of connections. **a**: Number of epochs required for a synfire chain to be learned with a 1 ms precision. **b**: Proportion of learning successes. **c**: Proportion of spurious firings.

#### 4.5 Catastrophic Interference

Catastrophic interference (for a review, see [7]) appears in neural networks when a particular learning experience is suddenly and completely destroyed by subsequent learning. A number of solutions to this problem have been proposed [6, 8]. This section explores catastrophic interference in the present network. In this simulation, involving 10 trials, the network had to learn sequentially 3 synfire chains. Thereafter, it was tested in order to see how much of each sequence it remembered. Results in Figure 8 show that the network remembered equally well all three synfire chains. These results seems to indicate that there is no catastrophic interference (the first learned chains are as well remembered as the last one). This lack of catastrophic interference could be due to the chaining rule in which only the most



reinforced connections indicate which nodes can participate in the next level of the chaining algorithm.

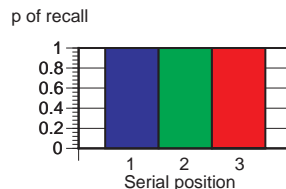


Figure 8: Three synfire chains were learned sequentially. The chart indicates an absence of catastrophic interference.

## 5. Conclusions

Human learning often involves relating two signals separated in time, or linking a signal, an action and a subsequent effect. These events are often separated in time, but nonetheless, humans can link them, thereby allowing them to accurately predict the right moment for a particular action.

There has been a recent surge in interest in the neurobiology of neuron spike timing. One of the major hypotheses concerning the timing behavior of neurons is synfire chains. Abeles and colleagues [1, 2, 17] have recorded neuron firing in the cortex of monkeys performing a task and have found stimulus dependent sequences of firing that were reproducible with a precision across trials of about 1 ms.

Synfire chains have been taken as a possible mechanism for representing relations between delayed events. This representation could enable the anticipation of an action or an effect following a signal. For example, a tennis player observing his opponent's serve, can anticipate the place where the ball will land and can prepare to return the ball. This is a learned ability. Consequently a set of synfire chain must be learnable and permit the linking of two events with a precise delay.

A learning algorithm based on a Hebbian learning rule has been presented in this paper. A number of simulations indicate that synfire chains can be learned, but that this learning is dependent on several factors, among them, the lag between the probe and the target stimuli, the size of the network (synfire chains require many nodes), the presence of long-term depression, and, finally the sparseness of connections between subnets. Furthermore, some presented results indicate that the learning algorithm presented here is not prone to catastrophic interference when it has to learn sequentially different synfire chains. Further studies are needed to explore how this learning algorithm could apply to more complex tasks and whether its behavior matches human data.

## Acknowledgements

This research was supported by the Belgian PAI Grant p4/19 and the European Commission grant HPRN-CT-1999-00065. Special thanks to Robert French for his assistance in the work presented here.

## References

1. Abeles, M. (1991). *Corticonics: Neural circuits of the cerebral cortex*. New-York: Cambridge University Press.
2. Abeles, M., Prut, Y., Bergman, H., Vaadia, E., & Aertsen, A. (1993). Integration, Synchronicity and Periodicity. In A. Aertsen (Ed.), *Brain Theory: Spatio-Temporal Aspects of Brain Function* (pp. 149-181). Amsterdam: Elsevier.
3. Bienenstock, E. (1995). A model of neocortex. *Network: Computation in Neural Systems*, 6, 179-224.
4. Calvin, W. H. (1983). A stone's throw and its launch window: timing precision and its implications. *Journal of Theoretical Biology*, 104, 121-135.
5. Engert, F., & Bonhoeffer, T. (1997). Synapse specificity of long-term potentiation breaks down at short distances. *Nature*, 388, 279-284.
6. French, R. M. (1997). Pseudo-recurrent connectionist networks: An approach to the 'sensitivity-stability' dilemma. *Connection Science*, 9, 353-379.
7. French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3, 128-135.
8. French, R.M., Ans, B., & Rousset, S. (2001). Pseudopatterns and dual-network memory models: Advantages and shortcomings. In R.M. French, & J.P. Sougné (Eds.) *Proceedings of the Sixth Neural Computation and Psychology Workshop: Evolution, Learning, and Development*. London: Springer-Verlag.
9. Hebb, D. O. (1949). *The Organization of Behavior*. New York: Wiley.
10. Hermann, M., Hertz, J.A., & Prügel-Bennet, A. (1995). Analysis of Synfire Chains. *Network: Computation in Neural Systems*, 6, 403-414.
11. Hertz, J.A., & Prügel-Bennet, A. (1996). Learning synfire chains by self organization. *Network: Computation in Neural Systems*, 7, 357-363.
12. Levy, N., & Hom, D. (1999). Distributed synchrony in a Hebbian cell assembly of spiking neurons. In *Advances in Neural Information Processing Systems 11*, Cambridge Ma, MIT Press.
13. Maass, W., & Bishop, C.M. (1999). *Pulsed Neural Networks*. Cambridge, Ma: MIT Press.
14. Markram, H., Lübke, J., Frotscher, M., & Sakmann, B. (1997). Regulation of Synaptic Efficacy by coincidence of Postsynaptic Aps and EPSPs. *Science*, 275, 213-215.
15. Markram, H., Gupta, A., Uziel, A., Wang, Y., & Tsodyks, M. (1998). Information processing with frequency-dependent synaptic connections. *Neurobiology of Learning and Memory*, 70, 101-112.
16. Munro, P., & Hernandez, G. (1999) LTD facilitates learning in a noisy environment. In *Advances in Neural Information Processing Systems 11*, Cambridge Ma, MIT Press.
17. Prut, Y., Vaadia, E., Bergman, H., Haalman, I., Slovin, H., & Abeles, M. (1998). Spatiotemporal structure of cortical activity: Properties and behavioral relevance. *Journal of Neurophysiology*, 79, 2857-2874.
18. Sougné, J. P. (1996). A Connectionist Model of Reflective Reasoning Using Temporal Properties of Node Firing. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science* (pp. 666-671). Mahwah, NJ: Lawrence Erlbaum Associates.
19. Sougné, J. P. (1998). Connectionism and the problem of multiple instantiation. *Trends in Cognitive Sciences*, 2, 183-189.
20. Sougné, J. P. (1999). INFERNET: A neurocomputational model of binding and inference. Doctoral dissertation, Université de Liège.
21. Sterratt, D. C. (1999). Is biological temporal learning rule compatible with learning synfire chains? *Proceedings of the Ninth International Conference on Artificial Neural Networks*.
22. Zhang, L.I., Tao, H.W., Holt, C.E., Harris, W., & Poo, M. (1998). A critical window for cooperation and competition among developing retinotectal synapses. *Nature*, 395, 37-44.