Learning by Stimulation Avoidance as a Primary Principle of Spiking Neural Networks Dynamics
Lana Sinapayen1, Atsushi Masumori1, Nathaniel Virgo2, and Takashi Ikegami1

Abstract
Practical implementation of the concept of reward has deep implications on what artificial-life based systems can learn and how they learn it. How can a system distinguish between useful behavior and harmful behavior? In this paper we implement reward/punishment as the removal/application of a stimulation to a recurrent spiking neural network with spike-timing dependent plasticity. This implementation embodies the concept of reward at the level of the neuron, making learning mechanisms ubiquitous to the network. We show that this low-level learning scales up to the network level: the network learns arbitrary spatio-temporal firing patterns purely by interacting with the environment, from a random initial state where virtually no knowledge is available. This approach yields fast, noise-robust results.

Introduction
Learning mechanisms discovered in biological neural networks are often translated into hypotheses that can be implemented and tested in artificial neural networks (Sejnowski et al., 1988). The dynamics of artificial neural networks have recently grown much closer to their biological counterpart, thanks to the introduction of biologically plausible models of spiking neurons (Izhikevich, 2004; Brette et al., 2007). These refined models could facilitate the exchange of concepts between artificial and biological networks. These exchanges lead to stronger theoretical understanding of the biological networks, and sometimes to real world applications in Artificial Intelligence (Kawato and Samejima, 2007; Cheng et al., 2007).

Rather than studying biological networks in vivo, networks of cortical neurons cultured ex vivo are often used to study biological learning mechanisms. Cultured networks are easier to study because they are composed of relatively few neurons, and are isolated from most external influences. Despite being much simpler than actual brains, they retain some important properties of in vivo neural networks (Canepari et al., 1997), among which: type and distribution of cells, high connectivity, spontaneous activity, diversity of firing patterns.

The research presented in this paper is based on the findings of Shahaf and Marom on ex vivo neural networks (Shahaf and Marom, 2001; Marom and Shahaf, 2002). They demonstrate that an ex vivo neural network can learn a desired behavior if stimulation removal is used as a reward, following this protocol: (1) Apply a low frequency, focal electrical stimulation to the network. (2) When the desired behavior appears, remove the stimulation. After several of these “training sessions”, the network learns to produce the desired behavior in response to the stimulation. In practice, the authors show that the network learns to produce spikes at a determined location and in a determined time window, in response to a stimulation applied at a different location in the network.

Despite the promising potential of the findings of Sahaf and Marom, it seems that the applications, limits and explanatory mechanisms of the method were not further studied in cultured or artificial neural networks. In Marom and Shahaf (2002), the authors cite two different classes of reward theory: “[M]apping the behavioral concept of reward to a neural entity that strengthens a subset of synapses based on past performance of the neural system” and “In contrast, the Stimulus Regulation Principle advocates that neural connectivity changes are due to the persistence of a driving stimulus [...]”. If the output of the system changes the driving stimulus by its removal, [...] the system is ‘frozen’ in its last conformation; no specific cellular and synaptic reward mechanism needs be postulated.” The authors invoke the Stimulus Regulation Principle to explain their results, but the actual mechanisms that could be implementing this principle in the biological network are not explored.

In this paper, we implement, extend and give some empirical justification to a new principle termed “Learning by Stimulation Avoidance” (LSA) using a small artificial neural network. We define the LSA principle as the reinforcement of behaviors leading to a decrease of stimulation, and the weakening of behaviors leading to an increase in stimulation. Implemented at the level of the neurons, LSA has the following properties: learning can be obtained selectively at a random location of the network if the stimulation is focal; LSA does not require the existence of a specific neural “rewarding module”; both population coding and time coding...
can be obtained as a result of LSA.

We postulate that Hebbian learning rules (Hebb, 1949) are not only sufficient to implement LSA in a spiking neural network, but that the use of spiking neural networks actually leads to the following emergent properties of Hebbian rules: exploration of the space of possible connections, stimuli differentiation, active strengthening of synapses after the execution of desired behavior, and synaptic pruning after the execution of undesirable behavior. LSA therefore relies on principles from both classes of reward theory cited by Marom, with the important addition of synaptic pruning.

Hebbian rules are often inaccurately summarized with the sentence “cells that fire together, wire together.” This sentence originates from the observation that in biological neural networks, when two neurons connected by a synapse fire at a short time interval from each other, their synaptic strength will change (Markram et al., 1997). But the sentence incorrectly pictures the actual interactions between two connected excitatory neurons:

- If neuron A fires just before neuron B, the weight of the synapse conducting signals from A to B will increase.
- If neuron A fires just after neuron B, the weight of the synapse conducting signals from A to B will decrease.

Therefore the actual dynamics of Hebbian rules depend on the delay between the two spikes and the timing of those spikes: cells that do fire together might well get unwired from each other. Furthermore, there are different mechanisms for inhibitory neurons leading to even more complex synaptic weight dynamics (Caporale and Dan, 2008). In this paper, we use Spike-Timing Dependent Plasticity (STDP, Caporale and Dan (2008); Song et al. (2000)) as an implementation of Hebbian rules for spiking networks. We focus on STDP between excitatory neurons exclusively. We demonstrate three hypotheses concerning LSA:

1. STDP alone is sufficient to realize LSA at the level of an individual synapse.
2. STDP-enabled neurons are able to react selectively to simultaneous simulations in several synapses.
3. Therefore, STDP-based LSA scales up to the level of an entire network.

Additionally, we show that this implementation is robust to noise, has a high success rate and is computationally fast.

**Design**

**Spike-Timing Dependent Plasticity**

We implement STDP as proposed by Bush et al. (2010). The equations and resulting weight variation are shown in Fig. 1, representing a situation where a neuron $N_A$ and a neuron $N_B$ fire within $s$ ms of each other. The weight of the synapse transmitting signals from $N_A$ to $N_B$ varies as:

$$w_t = w_{t-1} + \Delta w .$$  \hspace{1cm} (1)

We fix a maximum value to the weight: if $w > w_{\text{max}}$, $w$ is reset to $w_{\text{max}}$. In the experiments identified as such, we also apply a decay function to all the weights in the network. The decay function is applied at each iteration $t$ as:

$$\forall w_t, w_{t+1} = 0.9999995w_t .$$  \hspace{1cm} (2)

In the experiments, we simulate either minimal networks of 2 or 3 excitatory neurons with one output synapse each, or fully connected networks of 100 neurons (Recurrent Neural Network, or RNN). The RNN is composed of 20 inhibitory neurons and 80 excitatory neurons. Synapses are modeled by directed connections between two neurons (self-connections are proscribed). Every neuron is connected to each other neuron by a synapse of synaptic weight $w$. The output weights are positive for excitatory neurons and negative for inhibitory neurons. The weights are initially set to random values (uniform distribution, $0 > w > 5$ for connections from excitatory neurons and $-5 < w < 0$ for connections from inhibitory neurons). In both the minimal network experiments and the RNN-based experiments, the weight of each connection between excitatory neurons is updated at each iteration of the simulation according to the STDP rule. We do not apply STDP on input and output connections of inhibitory neurons.

**Network model**

For the implementation of the network, we use the spiking model of cortical neuron proposed by Izhikevich (Izhikevich, 2003). This model has two main advantages: it can be tuned to accurately reproduce the dynamics of different types of cortical neurons, and it is computationally ef-

---

Differential equations of the Izhikevich model

\[ v' = 0.04v^2 + 5v + 140 - u + I \]
\[ u' = a (bv - u) \]
\[ a = 0.02 \]
\[ b = 0.2 \]
\[ c = - 65 \text{ mV} \]
\[ d = 8 \]

The Izhikevich model has two main equations and a reset condition; the parameters and dynamics of regular spiking excitatory neurons and fast spiking inhibitory neurons are summarized in Fig. 2. We do not discuss here the details of the model; for the understanding of this paper it is sufficient to know that \( v \) (mV) is the membrane potential of the neuron, \( u \) the membrane recovery variable, \( I \) (mV) the sum of synaptic inputs and externally injected currents, and \( \Delta t \) (ms) is the time resolution. We fix the time resolution to \( \Delta t = 1 \text{ ms} \).

A specificity of the Izhikevich model is the role of “noisy thalamic inputs” (referred simply as “noise” in this paper). In this model, a noise variable \( m_i \) is added to the input of each neuron at each iteration. This noise regulates the global level of activity of the network: with no noise at all, the connections in a small network do not provide sufficient stimulation to lead to sustained activity. With too much noise, the network enters a state where all neurons are continuously spiking. We choose \( m \) as a variable with a zero-mean Gaussian distribution and a standard deviation of \( \sigma \). \( m \) takes a different value at each iteration and for each neuron \( N_i \).

When a neuron \( N_j \) fires, each other neuron has its input variable augmented by the weight of the input connection with \( N_j \). Therefore at each iteration, the input \( I_i^* \) received by a neuron \( N_i \) from all other neurons is

\[ I_i^* = \sum_{j=0}^{n} w_{j,i} \times f_j, \]  
\[ f_j = \begin{cases} 1, & \text{if } N_j \text{ is firing} \\ 0, & \text{otherwise} \end{cases} \]

with \( n \) the number of neurons in the network, \( w_{j,i} \) the weight of the connection from \( N_j \) to \( N_i \), and \( w_{0,i} = 0 \).

An external input \( e_i \), for example corresponding to a stimulation from an electrode, can be added to the input of a neuron. The total input in one neuron at each iteration is therefore

\[ I_i = I_i^* + e_i + m_i. \]

**Experimental protocol**

We perform two types of experiments. In the first type of experiment (Experiments 1 and 2), we study the basis of STDP-based LSA in a minimal network of 2 or 3 neurons. In the second type of experiment (Experiment 3), we show that LSA scales up to the level of a RNN with 100 neurons. The task that must be learned in the RNN experiments is synchronous population coding: we choose groups of neurons in the network and a firing pattern for each group. The combination of groups and firing patterns constitutes the “desired behavior” that the network must learn to exhibit in response to external stimulation.

The training protocol is as follows: (1) The experimenter defines the desired behavior; (2) External stimulation is applied until the desired behavior appears, or until a time delay is reached; (3) The stimulation is removed; (4) Step 2 and 3 constitute one training cycle; they are repeated until the delay between the beginning of the stimulation and the appearance of the desired behavior is systematically under a fixed value (behavior learned).

**Experiments and Results**

1. **STDP-based LSA in a single synapse**

We design the first experiment to study the weight variation in one synapse between two neurons (Fig. 3). We control the external stimulation \( e_0 \) in \( N_0 \) under 3 conditions summarized in Fig. 4: (a) Stop the stimulation if \( N_1 \) fires; (b) Start the stimulation if \( N_1 \) fires; (c) Stimulate \( N_0 \) whatever the behavior of \( N_1 \). The delay between two training cycles is 30 ms (except in (c) where the stimulation goes uninterupted). We set the stimulation as \( e_0 = 2 \text{ mV} \) and the noise standard deviation as \( \sigma = 10 \). These parameters are chosen rather arbitrarily as there is no network effect to take into consideration. The initial weight is \( w_{0,1} = 5 \).

At the beginning of the experiment, the synaptic weight is comparatively low and \( N_1 \) fires in reaction to both the firing of \( N_0 \) and the high random noise \( m_1 \). STDP is applied to the
Figure 3: Experimental setup: the minimal network counts 2 neurons and 1 synapse. Random noise $m$ is added as input to both neurons. An external stimulation $e_0$ is applied to $N_0$. The dynamics of $e_0$ depend on the experimental conditions.

Figure 4: The three conditions used in Experiment 1 summarized as a raster plot. The rectangles represent neuronal spikes. The external stimulation $e_0$ causes $N_0$ to fire rhythmically; firing of $N_0$ causes stimulation in $N_1$. In real conditions, the noise and synaptic weight variations cause less regular spiking.

Figure 5: Weight variation in one synapse depending on the effect of post-synaptic neuron firing. The principle of LSA is verified: behaviors conducting to stimulation avoidance are reinforced via synaptic strengthening or synaptic pruning. The default dynamics of STDP lead to weight strengthening in neutral conditions, behavior which can be partly avoided by applying a decay function.

In (c), $N_0$ is continuously stimulated. The synaptic weight increases slowly but continuously: as long as the firing of $N_1$ is not clearly the cause of the external stimulation of $N_0$, the connexion will be slowly strengthened. The slow increase of the weight, as opposed to stable variations around the initial value of 5, is explained by two factors. First, $N_0$ contributes to the stimulation in $N_1$. Therefore, $N_1$ if more likely to fire after $N_0$ fired: spikes of $N_1$ will on average be closer to the last spike $N_0$ than to its next spike. This leads to $w_{0,1}$ being reinforced; in return, this stronger weight causes smaller time delays between the spikes of the two neurons. This can potentially be exploited as an exploratory behavior, but in practice, in large networks it leads to a state of weight saturation where all neurons are constantly firing. One way to deal with the issue is to apply a decay function on the weights. In our network, noise is mainly responsible for the exploration process, so we apply the decay function to avoid weight saturation.

This simple experiment with a minimal network validates Hypothesis 1: STDP alone is sufficient to realize LSA at the level of an individual synapse. In a minimal network with a single synapse, the synapse is strengthened to reinforce post-synaptic firing if it leads to removal of pre-synaptic stimulation; the same synapse is pruned if post-synaptic firing causes pre-synaptic stimulation. Therefore STDP is sufficient to realize Learning by Stimulation Avoidance at the level of a single synapse. Additionally, by running experiments with different values of external stimulation $e_0$ and

noise \( m \), we find that the learning speed tends to decrease when the noise level or the stimulation level are decreased. For example, reducing the noise standard deviation to \( \sigma = 5 \) and the stimulation to \( e_0 = 1 \) leads to a weight of only 10 in the reinforced synapse after 100 000 ms, compared to a weight of 30 in Experiment 1. Both high noise and high stimulation values tend to increase the firing rate of \( N_1 \), which increases the learning speed. This leads to the paradoxical observation that noise increases the performance of the minimal network. Furthermore, the results still hold for extremely low signal to noise ratio (high noise, low stimulation). In the next experiment, we validate Hypothesis 2 and show that the minimal network is capable of selective learning.

2. Effect-based differentiation of stimuli

In Experiment 2, we add one neuron to the minimal network (Fig. 6). The noise standard deviation is reduced to \( \sigma = 5 \) to account for the increased stimulation in the network (due to both \( w_{2,1} \) and \( e_2 \)). The external stimulations \( e_0 \) and \( e_2 \) vary independently between 0 mV and 2 mV. \( N_0 \) is stimulated until \( N_1 \) fires, then \( e_0 \) is stopped for 30 ms. \( N_2 \) is stimulated by \( e_2 \) during those 30 ms. So the firing of \( N_1 \) causes external stimulation in \( N_2 \), but stops external stimulation in \( N_0 \). The network must tell apart these influences despite the noise: we expect \( w_{0,1} \) to increase as a realization of LSA, since firing of \( N_1 \) is beneficial to \( N_0 \) (it stops the external stimulation). Meanwhile, the firing of \( N_1 \) is detrimental to \( N_2 \), as it causes external stimulation: if LSA is realized, \( w_{2,1} \) should decrease. These are indeed the results of the experiment, as shown in Fig. 7: \( w_{0,1} \) (in blue) increases and \( w_{2,1} \) (in red) decreases at the same time. This result is explained by the fact that despite the noise, there are overall more spikes of \( N_0 \) just before spikes of \( N_1 \) than just after, leading through STDP to an increase in weight. Similarly, there are overall more spikes of \( N_2 \) just after spikes of \( N_1 \) than just before, leading to an decreasing weight.

We also perform a variant of this experiment where the stimulation in \( N_0 \) stops 5 ms after the firing of \( N_1 \) (instead of stopping instantly). Therefore not only the causality between the spikes of \( N_1 \) and the end of the stimulation is delayed, but additionally the stimulations in the two presynaptic neurons \( N_0 \) and \( N_2 \) overlap for 5 ms. Despite these additional difficulties, the results stay qualitatively the same as in the original experiment, with an increased learning speed (\( w_{2,1} \) reaches 0 at \( t \approx 40 \) 000 ms). The increase in speed is due to the increased firing rate of \( N_0 \) as a consequence of stimulation building up during the additional 5 ms.

These results validate Hypothesis 2: one neuron can receive simultaneous signals from two synapses and proceed to prune one while strengthening the other. Therefore the neuron will react differently to two stimulations with conflicting effects.

3. Synchronous population coding in a larger network

What works for one neuron may not work in a more complex network where all neurons influence each other, and where these influences are much more difficult to tell apart. In Experiment 3, we test the scalability of LSA in a network of 100 neurons, all connected to each other by synapses with initially random weights. Related works suggest that in large random networks, having around 20% of inhibitory neurons (as in the mammalian cortex) is important for the dynamics of the network (Izhikevich, 2003; Connors and Gutnick, 1990). Therefore we set 20 of the neurons (\( N_0 \) to \( N_{19} \)) as inhibitory neurons.

The experimental protocol is as follows: 10 neurons (\( N_{20} \) to \( N_{29} \)) are chosen as input neurons and externally stimulated (\( e = 0.8 \) mV). We monitor the activity in 20 output
neurons and define the task to be learned as follows: in a group of 10 output neurons (group A, $N_{30}$ to $N_{39}$), at least 4 neurons should fire simultaneously at time $t$ (resolution = 1 ms). In a different group of 10 neurons (group B, $N_{40}$ to $N_{49}$) and at the same time $t$, less than 4 neurons should be firing. The condition on group B ensures that we avoid a trivial solution where all 100 neurons synchronize their spikes. When the desired firing pattern is obtained from the outputs, we stop stimulating the input neurons. If the desired firing pattern is not obtained after 10 000 ms, the stimulation is also stopped. In both cases, the stimulation starts again 500 ms after stopping. The experiment lasts 600 000 ms. The noise standard deviation is $\sigma = 3$ and the maximum weight is $w_{\text{max}} = 15$.

Fig. 8 shows the learning curve of the network. At the beginning of the experiment, the desired output is obtained at random delays after starting the stimulation. This is an initialization phase where Izhikevich networks are subject to a few highly synchronized bursts (see also Fig. 9). After the initialization phase, there is an exploration phase where the stimulation is often stopped because the maximum stimulation time is reached, and seldom because the desired output was recorded. The exploration phase is followed by a learning phase, where the learning curve decreases steeply to short reaction times. Towards the end of the experiment, the desired output is obtained within short reaction times (less than 2 000 ms after the start of the stimulation). Fig. 9 shows the evolution of the network’s firing patterns. The initialization phase (a) is dominated by long, sparse, highly synchronized bursts involving all neurons. At the learning phase (b), these completely synchronized bursts have been replaced by more temporally distributed firing. After the desired behavior is learned in (c), the raster plot shows high global activity of the network, with short, strongly structured bursts.

We ran the same experiment on 10 additional randomly generated networks (10 min of simulation time). Out of these 10 networks, 7 reached a reaction time systematically inferior to 2 000 ms after only 5 min of simulation. For these networks, the average reaction time after learning ($t > 5 \text{ min}$) was 108 ms. The average reaction time of the remaining 3 networks was 1 056 ms after 5 min. All 10 networks showed an increase in the frequency of apparition of the desired output pattern and a decrease of the reaction time. To summarize, all networks successfully learned to exhibit the correct output pattern, each with an average reac-
Figure 10: Results of Experiment 3 with no inhibitory neurons in the network. The initialization phase is as long as in the original conditions, but the network is stuck in the exploration phase and never reaches the learning phase.

Figure 11: Results of Experiment 3 with $e = 0$ mV. The initialization phase is as long as in the original conditions, but with no way to differentiate the correct firing pattern from other firing patterns, the network has no proper exploration phase and does not reach the learning phase. The desired firing pattern stops being exhibited.

Discussion

Although there is no consensus on the definition of learning, we believe that Experiments 1 and 2 qualify as learning in its simplest expression: the minimal network’s behavior changes to adapt to external influences. One neuron learned to fire when the network was “rewarded” by stimulus removal, and the same neuron learned not to fire when the network was “punished” by the application of a stimulus. These experiments serve as simplifications to explain the results of Experiment 3. In Experiment 3, the network starts in a random state, with no knowledge about itself, the environment, or the task that should be learned. The input neurons are no different from the two types of output neurons or from the reservoir neurons; all these groups are randomly chosen by the experimenter. The rule governing the stimulation pattern is also unknown by the network. Furthermore, initially most spikes in the network are due to strong random noise (the noise’s standard deviation is more than three times as big as the stimulation). Noise is usually an issue in artificial networks, and in artificial systems in general. But here as in biological systems, the action of noise is mostly non-detrimental. It is even beneficial, as random firing is necessary to kick-start the learning process.

In true reinforcement learning fashion, the network learns entirely by interacting with the experimental environment and finds the correct output leading to stimulation avoidance. This “correct” output can be any combination of simultaneously firing/not firing neurons; the location and even the number of neurons implicated in the rule are unknown to the network. Despite the huge search space of all these possible combinations, and despite the impossibility of learning progressively as would be the case with a fitness function (in our case the output is either correct or incorrect, nothing in-between), the network correctly learns its ascribed task. Furthermore, the task is learned after less than 5 min of simulation time on average (about 3 min in real time using a rather slow laptop PC and a non-optimized programming language).
To summarize, our method allows temporal and population coding; it is robust to noise; it is simpler than most existing learning approaches for spiking networks and requires a smaller number of neurons; it is fast; it works with a wide range of parameters. In experiments yet unpublished, we find that the network can learn complex sequences of randomly chosen temporal patterns (polychronization) and can perform cost analysis over several stimulation sources. It also exhibits more classical forms of learning like Pavlovian conditioning.

On the other hand, one issue of this implementation is that it is not easily scalable. At the moment, either the noise level or the number of connections must be manually adjusted when working with networks of hundreds or thousands of neurons. This suggests that the network should be used in small modules or pathways loosely connected to each other. The true impact of all parameters (noise, stimulation, number of neurons and of connections, maximum weight) must be further studied in future works. Another issue not treated in this paper is the possibility to use other types of neurons with different dynamics.

**Conclusion**

The principle of Learning by Stimulation Avoidance yields promising results, and can be easily implemented in spiking networks. STDP might not be the only possible implementation of LSA, but associated to the Izhikevich network model, it leads to unique robustness and versatility. We believe that the implementation of LSA presented in this paper deserves to be submitted to more challenging tasks.

**Acknowledgments**

This work was supported by Grant-in-Aid for Scientific Research (Studies on Homeo-Dynamics with Cultivated Neural Circuits and Embodied Artificial Neural Networks; 24300080).

**References**


