

A comparison of learning abilities of spiking networks with different spike timing-dependent plasticity forms

Alexander Sboev^{1,2}, Danila Vlasov¹, Alexey Serenko², Roman Rybka²,
Ivan Moloshnikov²

¹MEPhI National Research Nuclear University, Moscow, Russia

²NRC “Kurchatov Institute”, Moscow, Russia

E-mail: vfked0d@gmail.com, serenko@phystech.edu, rybkarb@gmail.com,
sag111@mail.ru, ivan-rus@yandex.ru

Abstract. A study of possibility to model the learning process on base of different forms of timing-dependent plasticity (STDP) was performed. It is shown that the learning ability depends on the choice of spike pairing scheme and the type of input signal used for learning. The comparison of performance of several STDP rules along with several neuron models (leaky integrate-and-fire, static, Izhikevich and Hodgkin-Huxley) was carried out using the NEST simulator. The combinations of input signal and STDP spike pairing scheme, which demonstrate the best learning abilities, were extracted.

1. Introduction

Despite the great amount of devoted research, the problem of development for spiking network a learning mechanism, which is biologically founded and practically effective, is far from resolution. There is a number of approaches [3, 4] to make spiking neural networks learn to respond with the desired output to a given input with the help of different variations of Hebbian-inspired learning. This task involves two problems:

- firstly, what input-output transformations a network can in principle implement with some stable values of its synaptic weights;
- secondly, whether the network can learn the desired transformation with the chosen learning protocol.

Obviously, a neuron cannot produce any output spike train in response to arbitrary input spike trains. A less obvious problem, discussed in the section 4.5, is that the desired input-output transformation may be possible but require synaptic weights that are unstable with the chosen synaptic plasticity form.

Provided that the desired input-output transformation can be performed by the neuron with some stable synaptic weights, in [1] a supervised learning protocol is suggested. It is based on clamping the neuron output to desired, and the ability of any transformation, meeting the requirements stated above, to be learnt by this protocol was proven for Leaky Integrate-and-Fire neuron model for the case of all synapses receiving Poisson input trains with same mean frequency.



In this study we show that in case of different neuron models and more complex inputs learning is also possible (Section 4.2). However, the learnability is heavily affected by the choice of the scheme of pairing spikes in the STDP weight change rule (Section 4.1).

The reason in some cases is that the protocol of forcing the neuron to fire in desired moments by stimulating it with current impulses may, firstly, sometimes fail to cause a desired spike, and, secondly, not prevent the neuron from undesired spikes, caused by incoming synaptic currents instead of teacher stimulation. To prove this a learning protocol without a neuron, in which the weight change rule is just applied to the input trains and the desired output, is considered in Section 4.3, and it provides more successful learning.

2. Neuron and synapse models

In the Leaky Integrate-and-Fire neuron model the membrane potential V changes with time as following:

$$\frac{dV}{dt} = \frac{-(V(t) - V_{resting})}{\tau_m} + \frac{I_{syn}(t)}{C_m} + \frac{I_{ext}}{C_m}, \quad (1)$$

When the potential reaches the threshold value V_{th} , the neuron fires a spike, then its potential is instantaneously dropped to the value V_{reset} , and during the refractory period τ_{ref} the synaptic current has no effect on the potential. $V_{rest} = 0$, V_{th} was chosen to be 15.3 mV in all sections unless other stated, the other LIF neuron parameters were taken as in [1].

We also used the Hodgkin-Huxley [5] and Izhikevich [6] neuron models, taking the constants same as in the original works cited here, and static adder as a neuron, which simply accumulates weights of synapses from which it receives spikes, and when the accumulated value reaches the threshold V_{th} , the sum is dropped to zero and the neuron fires a spike.

As the synapse model we used postsynaptic current of exponentially decaying form along with the Maass-Markram short-term plasticity [8], taking all parameters as in [1].

2.1. Spike-timing-dependent plasticity

STDP is a biologically inspired long-term plasticity model [7], in which each synapse is given a weight $0 \leq w \leq w_{max}$, characterizing its strength, and its change depends on the exact moments t_{pre} of presynaptic spikes and t_{post} of postsynaptic spikes:

$$\Delta w = \begin{cases} -W_- \cdot \frac{w}{w_{max}}^{\mu_-} \cdot \exp\left(-\frac{t_{pre} - t_{post}}{\tau_-}\right), & \text{if } t_{pre} - t_{post} > 0, \\ W_+ \cdot \left(1 - \frac{w}{w_{max}}\right)^{\mu_+} \cdot \exp\left(-\frac{t_{post} - t_{pre}}{\tau_+}\right), & \text{if } t_{pre} - t_{post} < 0; \end{cases} \quad (2)$$

where $W_+ = 0.3$ and $W_- = 1.035 \cdot W_+$. The rule with $\mu_+ = \mu_- = 0$ is called additive STDP, with $\mu = \mu_- = 1$ — multiplicative, the one with $\mu_+ = 0$ and $\mu_- = 1$ is the van Rossum rule; intermediate values $0 < \mu < 1$ are also possible.

In case of additive STDP the auxiliary clause is added to prevent the weight from falling below zero or exceeding the maximum value w_{max} :

$$\text{if } w + \Delta w > w_{max} \text{ or } w + \Delta w < 0, \text{ then } \Delta w = 0.$$

2.2. Spike pairing schemes in the STDP rule

An important part of STDP rule is the scheme of pairing pre-and postsynaptic spikes when evaluating weight change according to the rule (2). Three nearest-neighbour schemes [7] are shown on fig. 1: symmetric A, presynaptic-centered B and reduced symmetric C. Black tics denote moments of spikes, and gray lines mean taking into account that pair of spikes in the STDP weight update rule.

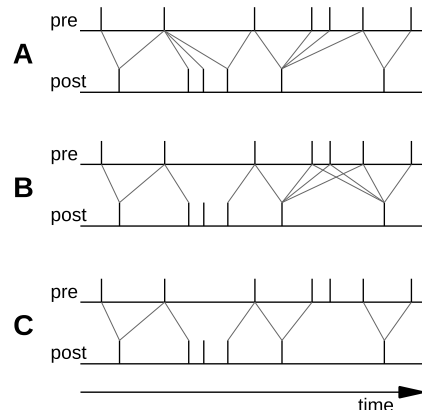


Figure 1. Spike pairing schemes: A – symmetric, B – presynaptic-centered, C – reduced symmetric. Redrawn from [7]

3. Experiment technique

3.1. The learning protocol

The following protocol suggested in [1] is to force the synaptic weights of a neuron to converge to the target weights:

(i) Obtaining the teacher signal

The neuron's weights are set equal to the target, STDP is disabled, and the neuron's response to the input trains is recorded as the desired output.

(ii) Learning

The neuron's weights are set random (but about 4 times smaller than the target ones), STDP is turned on, and the same input trains are given to the neuron's incoming synapses. During this the neuron is stimulated by the teacher signal, obtained from the desired output train by replacing spikes with 0.2-ms-duration current impulses of 2 mA (which is thousands of orders more than typical magnitude of synaptic current).

Another protocol, further called Learning without neuron, is not to use any neuron at all on the Learning stage, just to calculate weights change by the STDP rule using the input trains and the desired output train.

3.2. The experiment configuration

The one neuron had 90 excitatory and 10 inhibitory synapses. The maximum synaptic weights in the STDP rule (2) were chosen from $N(54, 10.8)$, values less than 21.6 and more than 86.4 being replaced by 21.6 and 86.4 correspondingly. The target weights \vec{W}_{target} were chosen to obey bimodal distribution: half of them were equal to zero, and the other half – to their maximum values; because such weights distribution is known [7] to be a steady state settled by the additive STDP rule.

For the implementation of neuron and synapse models we used the NEST simulator [9], modified by the authors of [2] to implement different spike pairing schemes.

4. Results

4.1. Impact of the spike pairing scheme on the learning performance

Legeistein et al. in [1] used not a nearest-neighbour, but the all-to-all spike pairing scheme, and used 20-Hz Poisson spike trains as input. In [2] the same experiment was conducted with three nearest-neighbour schemes shown on Fig. 1, and the ability of weights to converge to the target was shown to depend on the choice of spike pairing scheme.

As a measure for learning performance the deviation β between current and target weights was used:

$$\beta(t) = \frac{\sum_{i=1}^{90} |W^i(t) - W_{target}^i|}{\sum_{i=1}^{90} W_{target}^i}.$$

Table 1. β after learning with different spike pairing schemes in STDP rule and different input types

STDP scheme	uniform	normal	Poisson
all-to-all		0.11	0.11
A	1	1	0.8
B	0.10	0.10	0.08
C	0.08	0.08	0.05

means that the interspike intervals in the input trains were normally distributed, and uniform input means that on every simulation step the neuron had a fixed probability of receiving a spike from that input.

The scheme C demonstrates the best learning performance, so it is the one used for the rest of the paper.

So, the closer β is to 0, the more successful the learning is.

In Table 1 β after 3,000 seconds of training LIF neuron having additive STDP with different spike pairing schemes and receiving input of 20-Hz-mean trains to all synapses is shown, averaged over 5 independent simulations. The neuron threshold V_{th} was chosen about 15.6 mV, so that the mean output frequency was about 25 Hz. Normal input

4.2. Possibility of learning with different types of input signals and different neuron models

As can be seen from Table 2, different neuron models can learn target weights not only in case all synapses of the neuron receive trains of one constant mean frequency, but also if frequency of each input train changes every 100 s. The amplitude of current impulses the neuron was stimulated with on the Learning stage was 2 mA. The static neuron threshold was 15300, and, since such type of a neuron cannot be stimulated with current, the Learning stage in this case was performed according to the Without a neuron protocol.

Table 2. β after training different neuron models with different input types

Neuron model	10 Hz		10, then 60 Hz		10, 30, 50, 100 Hz	
	normal	poisson	normal	poisson	normal	poisson
LIF neuron	0.09	0.07	0.04	0.04	0.04	0.05
Hodgkin-Huxley neuron	0.10	0.07	0.03	0.04	0.06	0.11
Static neuron	0.03	0.03	0.05	0.08	0.19	0.35

4.3. Learning without a neuron, based only on input and output signals and STDP rule

Slightly less successful learning is observed if different synapses receive inputs of different mean frequencies. On Fig. 2 is an example of learning with input that causes rather poor weights convergence: half of synapses received Poisson trains with mean frequency of 30 Hz, and the other half — 10 Hz. It can be seen that increasing the neuron threshold on the Learning stage by 1 mV compared to the Obtaining teacher signal stage improved the weights convergence. Further increase in threshold does not provide more improvement. So the reason was supposed to be in avoiding unforced spikes, i.e. caused by the incoming synaptic currents instead of teacher impulses, and therefore differing from the desired output.

Another fact we found is that choosing the magnitude of the teacher current impulses to be 1 mA instead of 2 mA impairs learning, but increasing it above 2 mA does not improve the learning more. The reason was supposed to be that high teacher current avoids cases when a teacher impulse does not force the neuron to fire.

To confirm the explanations of these two facts we introduced the learning protocol without neuron, using just the inputs and the desired output. In the context of the two facts being discussed in this section, such a protocol provides the idealised way of learning, with no difference between the postsynaptic train and the desired output. Fig. 2 shows that learning without a

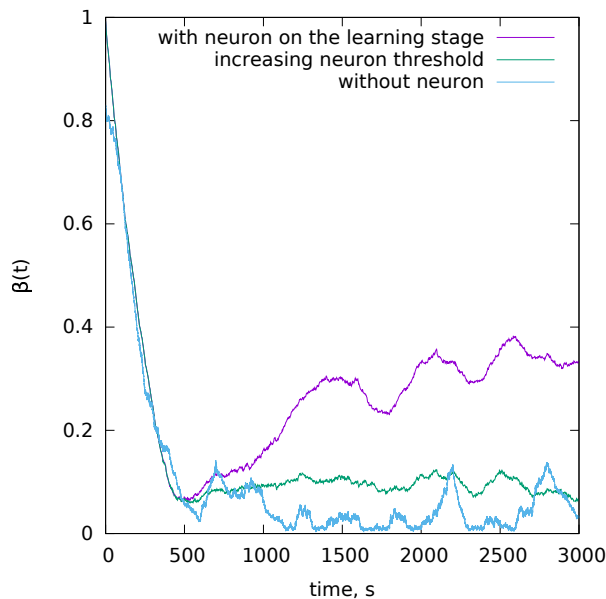


Figure 2. $\beta(t)$ during training LIF neuron with additive STDP. In this case the neuron had 100 excitatory and 90 inhibitory synapses; half of excitatory and all inhibitory synapses received Poisson trains with mean frequency of 30 Hz, and another half of excitatory synapses – of 10 Hz. The LIF neuron threshold was 15.3 mV on both stages (violet), then increased to 16.3 on the Learning stage (green), then learning without a neuron was performed (blue)

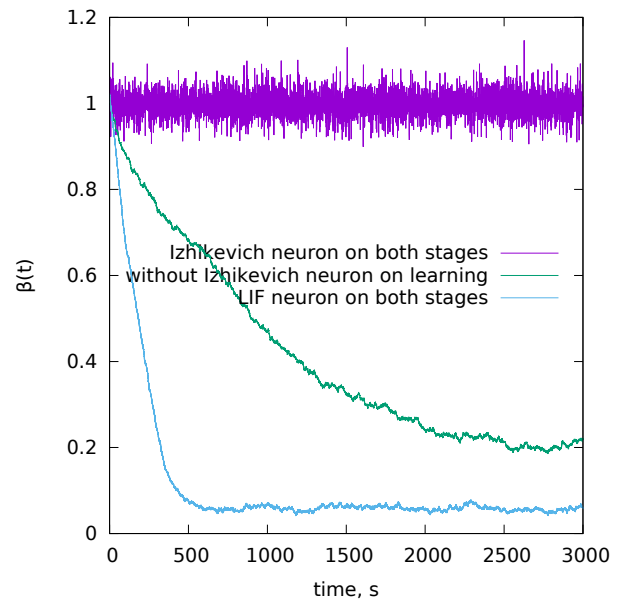


Figure 3. $\beta(t)$ during learning when Izhikevich neuron was used on both stages; when Izhikevich neuron was used on the Getting teacher signal stage, and learning was performed without neuron; and, for comparison, when LIF neuron was used on both stages. Both neurons were producing output with roughly same mean frequency

neuron is even more successful than the one with neuron and with increasing its threshold on the learning stage.

As a result, we conclude that differences in the learning performance of the protocols with and without the neuron are caused by the disadvantages of the protocol of presenting the teacher signal to the neuron in the form of current impulses, while, if the learning with some input and output trains was unsuccessful even without neuron, the reason would be that such input-output transformation cannot be learnt ever.

4.4. Learning with the Izhikevich neuron

No learning is possible using the Izhikevich neuron model on both Getting reinforcement signal and Learning stages, but in the case of using the protocol without a neuron with teacher signal obtained with Izhikevich neuron learning does take place (Fig. 3). So, the protocol of stimulating the neuron with current impulses does not work with the Izhikevich neuron, though output trains it produces can be used as desired during learning.

In case of additive STDP used in [1] and in this study (except the current section), only weight values of zero and maximum are equilibrium points, so target weights must obey the bimodal distribution in order to be stable. It must restrict the variety of possible input-output transformations.

4.5. Impossibility of learning with non-additive STDP forms

If learning with multiplicative STDP, i. e. with $\mu_+ = \mu_- = 1$ in weight change rule (2), was possible, intermediate values of weights could be reached during learning. However, in case of multiplicative STDP weights do not converge to the target (see Fig. 4). Increasing the dependency of weight change Δw on w makes learning less successful. Choosing $\mu_+ = 0.12$ and $\mu_- = 0.1236$ (as in [1]) leads to rather poor weights convergence, despite this values are still below the critical value of $\mu = 0.023$ exceeding which the steady state of weights from bimodal becomes [7] unimodal. Therefore, making intermediate values able to be reached as a result of learning still remains a question.

5. Conclusion

An output train can be used as desired during learning with some input only if it was obtained in response to that input, but regardless of the neuron model it was obtained with. In other words, output must be somehow correlated with the input. Nevertheless, learning may be impossible even with desired output correlated with input, if such input-output transformation can only be realised by a neuron with non-bimodal weights.

During learning by stimulating the neuron to fire at desired moments, deviations of the neuron's spike times from the desired output severely impairs learning performance compared to the case when output during learning exactly satisfies the desired one, as in the protocol without a neuron. The reason is that in STDP, like in any Hebbian learning rule, a postsynaptic spike can only lead to increasing synaptic weights due to "pre-before-post" part of the rule, regardless of whether it is desired or not. To provide the mechanism of penalizing synapses for causing undesired spikes, in most works [3, 4] some anti-Hebbian clauses are introduced, while their biological plausibility has not yet been proved. Development of both efficient and biologically plausible learning protocol still remains an unsolved problem.

References

- [1] R. Legenstein, C. Naeger, W. Maass, "What Can a Neuron Learn with Spike-Timing-Dependent Plasticity?", *Neural Computation*, vol. 17, 2005, pp. 2337–2382.
- [2] K. Kukin, A. Shoen, "Comparison of Learning Methods for Spiking Neural Networks", *Optical Memory and Neural Networks (Information Optics)*, vol. 24, 2015, No. 2, pp. 123–129 (2015)
- [3] R. Gütig, H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions" *Nat. Neurosci.* 9 (3), 2006, pp. 420–428
- [4] S. Mitra, S. Fusi, G. Indiveri, "Real-Time classification of complex patterns using spike-based learning in neuromorphic VLSI", *IEEE Transactions on biomedical circuits and systems*, vol. 3, 2009, No. 1
- [5] A. L. Hodgkin, A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve", *J. Physiol.*, vol. 117, 1952, pp. 500–544
- [6] Eugene M. Izhikevich, "Simple model of spiking neurons", *IEEE Transactions on Neural Networks*, vol. 14, No. 6, Nov. 2003
- [7] A. Morrison, M. Diesmann, W. Gerstner, "Phenomenological models of synaptic plasticity based on spike timing", *Biol. Cybern.*, vol. 98, 2008, pp. 459–478.
- [8] W. Maass, H. Markram, "Synapses as dynamic memory buffers", *Neural Networks*, vol. 15, 2002, pp. 155–161
- [9] Gewaltig M.-O., Diesmann M., "NEST (Neural Simulation Tool)", *Scholarpedia* 2(4), 2007, p. 1430.

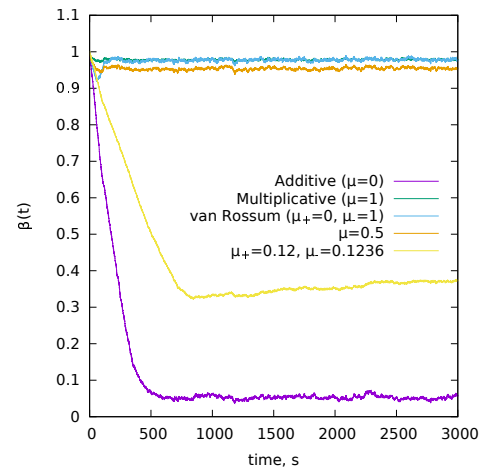


Figure 4. $\beta(t)$ during training without neuron with 20 Hz Poisson trains as inputs with different μ_+ and μ_- in the STDP weight update rule: additive $\mu_+ = \mu_- = 0$, multiplicative $\mu_+ = \mu_- = 1$, van Rossum $\mu_+ = 0$ and $\mu_- = 1$, The desired output was obtained with LIF neuron having threshold of 15.3 mV.