Encoding sequences of spikes in spiking neural networks through reinforcement learning

Filip Ponulak[1,2] and Stefan Rotter[1]

[1] Bernstein Center for Computational Neuroscience,
Albert-Ludwigs University Freiburg, Germany

[2] Institute of Control and Information Engineering,
Poznań University of Technology, Poland

{ponulak,rotter}@bcf.uni-freiburg.de

Introduction

We address the question of how networks of biologically plausible spiking neuron models can learn target transformations of input-to-output signals encoded in precisely timed patterns of spikes.

Gradient-based learning algorithms, like backpropagation, successfully solve this problem for networks of rate-based units. However, explicit evaluation of gradient in spiking networks is difficult due to their discontinuous dynamics. Indirect approaches or special simplifications must be assumed to deal with this problem [3].

Here we introduce an online reinforcement learning algorithm, which consistently modifies all synaptic weights in multi-layer or recurrent spiking neural networks without the requirement to calculate the gradient. We demonstrate that with this approach networks can learn to reproduce target sequences of spikes in response to the corresponding input patterns.

Algorithm

The algorithm is defined as follows. For any synaptic connection from neuron i to neuron j the synaptic efficacy $w_{ji}(t)$ is updated according to the formula:

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{ji}(t) \;=\; \alpha\, r_j(t)\, \overline{S_i(t)}, \qquad (1)$$

where α is the learning rate, $r_j(t)$ is the reward signal assigned to neuron j and $\overline{S_i(t)}$ is the eligibility trace of neuron i. We define the eligibility trace as:

$$\overline{S_i(t)} = a + \int_0^\infty A \exp(-s/\tau)\, S_i(t-s)\, \mathrm{d}s, \qquad (2)$$

with the constant parameters $a, A, \tau \in \mathbb{R}^+$ and with $S_i(t)$ being a spike train generated by neuron i. A spike train is defined as: $S(t) = \sum_f \delta(t - t^f)$, where t^f is a firing time, $f = 1, 2, ...$ is the label of the spike and $\delta(n)$ is the Dirac function.

Distinct definitions of a reward function are used for output and hidden neurons.

For each output unit o a positive reward $r_o(t) = +1$ is assigned to its synaptic inputs at every time t^d assumed to be the target firing time for the neuron o; a negative reward $r_o(t) = -1$ is applied whenever the neuron generates a spike; no reward is given $r_o(t) = 0$ at all other times. Formally this can be expressed as:

$$r_o(t) = \left(S_o^d(t) - S_o(t)\right), \qquad (3)$$

where $S_o^d(t)$ and $S_o(t)$ are the target and output spike trains of neuron o, respectively.

For every hidden neuron h the reward $r_h(t)$ is the sum of rewards assigned to the particular output neurons, i.e.:

$$r_h(t) = \sum_{o=1}^{n} r_o(t), \qquad (4)$$

where n is the total number of output neurons in the network.

The algorithm given by Eqs.(1-4) can be applied both to excitatory and inhibitory connections provided that inhibitory synapses are represented by negative weight values.

In order to ensure that the activity of the output neurons converges to the target patterns despite the continuous rearrangement of the synaptic inputs, the dynamics of weight changes in the output layer should be much faster then in the hidden layer. This is achieved by selecting α several times smaller for the hidden neurons as compared to the output neurons.

Results

Typical results of learning are presented in Fig.1. A network of leaky-integrate-and-fire neurons with an input layer (200 inputs), a hidden layer (400 neurons) and a single output neuron, is trained to reproduce a target Poisson spike train in response to randomly generated input signals (here we assume that the particular inputs fire once at a fixed time chosen randomly from a uniform distribution between 0 and 200ms.).

Each network input is connected randomly to around 25% of neurons in the hidden layer and all hidden neurons project on the output neuron. Around 20% of all synaptic connections in the network are assumed inhibitory.

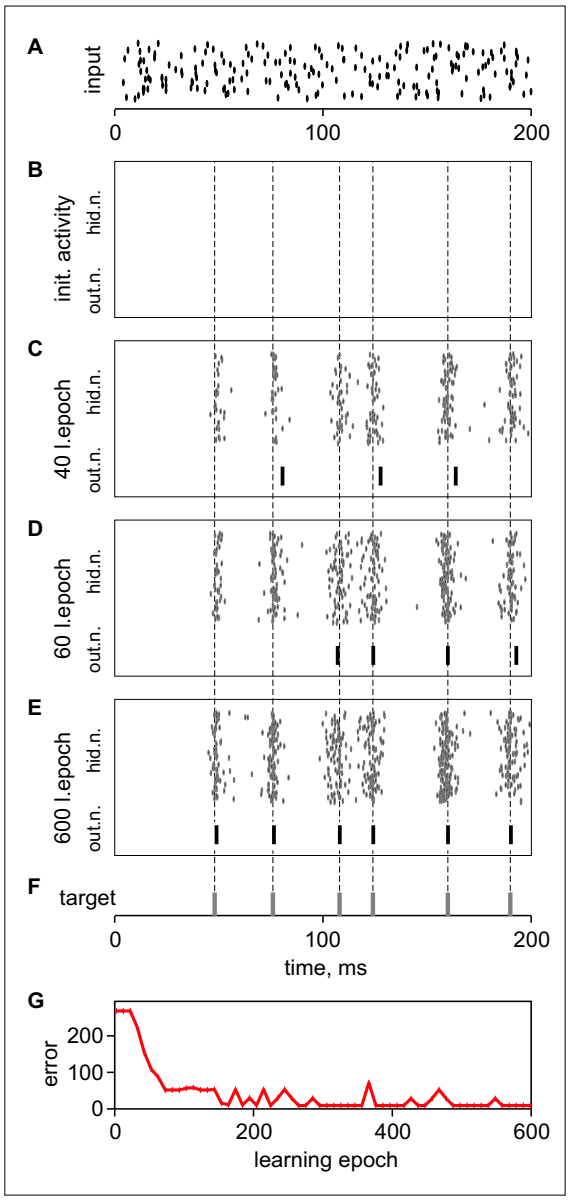

Figure 1: Illustration of the learning process. Spiking neural network is trained to generate the target sequence of spikes (F) in response to the given spatio-temporal input pattern (A). Network activity (hid.n.) and network output (out.n) are illustrated after selected learning epochs (B-E). (G) Learning error is consistently decreased in the consecutive epochs to reach the value close to 0 after around 300 steps.

Synaptic weights are initialized (according to a Gaussian distribution) such that the inputs do not initially evoke activity in the hidden layer (Fig.1.B). This is to demonstrate the ability of our algorithm to deal with silent neurons (as opposite to the backprop- or STDP-based algorithms).

The learning progress is illustrated after 40, 60 and 600 learning epochs (Fig.1.C-E, respectively). In the consecutive graphs we observe a slowly increasing activity in the hidden layer. However, only those hidden neurons start to fire which receive projections from the inputs that are active shortly before the target firing times. Consequently the neural activity in the hidden layer increases only around the target firing times. This activity gives rise to the spikes in the output neuron. As the learning continues the output spikes move towards the target times. Starting from around 300-th learning epoch the target pattern is almost perfectly reproduced at the network output with the spike-train correlation equal to 0.95.

Discussion

The algorithm introduced here extends our previous learning method, called ReSuMe, designed for single spiking neurons (or single-layer spiking networks) only [4, 5]. By enabling consistent modifications of all synaptic weights in a network, the new algorithm improves learning capability and memory capacity of the network as compared to ReSuMe.

Our approach possesses also several advantages over another, recently extensively explored, algorithm known as reward-modulated spike-timing-dependent-plasticity (RM-STDP) [1, 2].

That is, our algorithm solves the problem of training networks with silent neurons (as illustrated in the results section); the definition of a reward signal proposed here enables to effectively learn temporal sequences of spikes and ensures that the learning process reaches a stable fixed point whenever the output spike train matches the target pattern, which is not the case for the RM-STDP models considered so far.

Possible directions for future work include implementation of the algorithm in the actor-critic architecture or extension of the learning rules by additional terms ensuring further improvement of the learning performance.

Acknowledgement

This work was partially supported by the German Federal Ministry of Education and Research (grant 01GQ0420 to BCCN Freiburg).

References

[1] R. Florian. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural Computation*, 19(6):1468–1502, 2007.

[2] E. M. Izhikevich. Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling. *Cerebral Cortex*, 17(10):2443–2452, 2007.

[3] A. Kasiński and F. Ponulak. Comparison of Supervised Learning Methods for Spike Time Coding in Spiking Neural Networks. *Int. J.of Applied Mathematics and Computer Science*, 16(1):101–113, 2006.

[4] F. Ponulak. ReSuMe - new supervised learning method for Spiking Neural Networks. Technical Report, Poznan University of Technology, 2005. Available from: `http://d1.cie.put.poznan.pl/~fp/`.

[5] F. Ponulak and A. Kasiński. Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification and Spike-Shifting. *Neural Computation (submitted)*, 2008.