ESANN'2006 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2006, d-side publi., ISBN 2-930307-06-4.

Generalization Properties of Spiking Neurons Trained with ReSuMe Method

Filip Ponulak and Andrzej Kasiński *

Poznań University of Technology, Institute of Control and Information Engineering
Poznań 60-965, ul.Piotrowo 3a, Poland

Abstract. In this paper we demonstrate the generalization property of spiking neurons trained with ReSuMe method. We show in a set of experiments that the learning neuron can approximate the input-output transformations defined by another - reference neuron with a high precision and that the learning process converges very quickly. We discuss the relationship between the neuron I/O properties and the weight distribution of its input connections. Finally, we discuss the conditions under which the neuron can approximate some given I/O transformations.

1 Introduction

Ability of learning from examples and the generalization property of neural networks are the main reasons for the wide interest of researchers in the field of neural computation. These properties of neural networks are crucial for their applications in such tasks as function approximation, classification, identification, modelling or control [1].

In this article we focus our attention on the generalization property of spiking neurons trained with ReSuMe [2]. Spiking Neural Networks (SNN) represent a special class of artificial neural networks in which information is carried by the timing of particular spikes [3]. Thus SNN are particularly suitable to process information encoded in time. It has been demonstrated that spiking neurons are computationally more powerful than other neural units [4].

ReSuMe is a novel, efficient method of learning in SNN. ReSuMe takes advantage of the spike-based Hebbian processes [5] and integrates them with a novel concept of remote supervision introduced in [2]. The method enables supervised learning while still inheriting interesting properties of unsupervised Hebbian approach, i.e. the locality in time and space, simplicity and the suitability for online processing. On the other hand, ReSuMe avoids drawbacks of the Hebbian- and, so called, supervised-Hebbian methods (see e.g. methods discussed in [6, 7]).

Previously we have demonstrated that ReSuMe can effectively learn complex temporal and spatio-temporal patterns of spikes with the desired accuracy [8]. Here we extend these investigations and discuss a set of experiments, which show that ReSuMe enables learning of a wide class of transformations, understood as mappings of input to output spike trains. It appears that the spiking neurons trained with ReSuMe are able to transform inputs to the desired output signals

*The work was partially supported by the State Committee for Scientific Research, project 1445/T11/2004/27.

generated by the given reference object, for the patterns not used during the training. In that sense the neurons demonstrate the generalization property.

2 Methods

In our experiments we investigated the deterministic Leaky-Integrate-and-Fire neuron models and the static, reliable synapses [3]. To simulate the spiking structures we used CSIM: A Neural Circuit SIMulator [9]. During the training phase the synaptic connections were modified according to ReSuMe learning rules, while during the testing phase all synaptic weights have been kept fixed.

In the ReSuMe method the efficacy w of any synaptic connection between a presynaptic neuron n^{in} and a postsynaptic neuron n^l is modified according to the following rule:

$$\frac{\mathrm{d}}{\mathrm{d}t}w(t) \;=\; \left[S^d(t) - S^l(t)\right]\left[a + \int_0^\infty W(s)\, S^{in}(t-s)\;\mathrm{d}s\right], \tag{1}$$

where $S^d(t)$, $S^{in}(t)$ and $S^l(t)$ are target, pre- and postsynaptic spike trains, respectively. The spike trains are defined here by the sums of the firing times [3]. It is assumed that the target signal $S^d(t)$ is produced by the output of some reference neuron n^d. The parameter a expresses the amplitude of the non-correlation contribution to the total weight change, while the convolution function represents the Hebbian-like modifications of w. The integral kernel $W(s)$ is known as a learning window defined over a time delay s between the spikes occurring at the pre- and postsynaptic sites [3, 2]. For the excitatory synapses the term a is positive, and the learning window $W(s)$ has the shape similar as in STDP. For the inhibitory synapses a is negative and $W(s)$ is defined similarly as in the anti-STDP rules. For the complete introduction to ReSuMe we refer readers to [2].

After each learning session k we computed the correlation index $C(k)$ in order to quantitatively measure the quality of learning [7]. $C(k)$ is defined as a cross-correlation of the analog signals obtained from the $S^d(t)$ and $S^l(t)$ spike trains in the low-pass filtering operation. In the experiments reported here the correlation was computed for the time segments of length 5 seconds. The parameter $C(k)$ is sensitive not only to the spike missing or to spurious firings, but also to the spike-time shifts.

3 Results

In a set of experiments a single learning neuron n^l was trained to approximate some given I/O transformations defined by a reference neuron n^d. We used a spiking neuron as a reference object to ensure that any transformation produced by this object could be potentially reproduced by n^l (we refer to [7] for details).

It was assumed that n^d and n^l were characterized by similar dynamics, but the neurons differed in the initial weight distributions ($\bar{w}^d$, $\bar{w}^l$, respectively) in their inputs. In such a case the discrepancy of I/O properties of n^d with respect to n^l were uniquely determined by the difference between $\bar{w}^d$ and $\bar{w}^l$.

In the experiments each neuron received the common set of input signals through its m synapses. We randomly generated the weight distribution individually for n^l and for n^d, however we ensured that the number of excitatory/inhibitory synapses were the same in n^l as in n^d (Fig.1.D).

In the first experiment we trained n^l with 5 spike patterns of the length 100 seconds. Each pattern was characterized by different rate of spikes (26 up to 42 spikes per second). We used a randomly generated recurrent spiking network to transform each individual input pattern into a set of $m = 30$ spike trains driving both n^l and n^d. The response of n^d to this patterns was used as a target signal for n^l. During training the synaptic weights $\bar{w}^l$ at n^l were modified according to eq.(1). The analysis of the correlation index computed in the consecutive learning sessions for every spike pattern (Fig.1.A) shows that the mean correlation value increased significantly, already after two learning sessions, from $C(0) = 0.29$ (computed before the training) to $C(2) = 0.91$. This result reveals that we obtained a good approximation of the target signal. It also demonstrates that the learning process converges quickly. We continued with learning for the next 10 learning sessions, however we did not observe the significant changes in the C values. This confirms the stability of the solution.

After the training, the I/O properties of n^l were tested on 160 spike patterns. The patterns were generated by the recurrent spiking network. For each generated pattern the network was characterized by the different initial network states and the different spike-rates of the network input signals (the rates ranged from 20 to 50 spikes/second). The correlation C was computed for every set of testing patterns (Fig.1.B, grey dots) and compared to the values of C obtained for the signals used during learning (black dots). The results show that the quality of approximation observed in the validation set is comparable to that obtained in the learning set. This demonstrates that the learning neuron has properly generalized the learned I/O relationship (Fig.1.C).

Such a good generalization ability can be explained in the context of the relationship between the I/O properties of a neuron and its input weight distribution. If we compare $\bar{w}^d$ and $\bar{w}^l$ before training (Fig.1.D, middle), we see that these distributions differ significantly. This results in a low correlation between the signals produced initially by n^l and n^d. Contrary, after the training the distribution $\bar{w}^l$ becomes very close to $\bar{w}^d$ (Fig.1.D, right). Thus the trained neuron n^l and the target neuron n^d are supposed to exhibit similar I/O properties, independently on the used input signals.

We tested our approach on a number of variants with the different number of inputs, different input signal distributions and lengths as well as with the different initial weight distributions (an example is presented in Fig.2). In each case we obtained positive generalization property of the learning neuron.

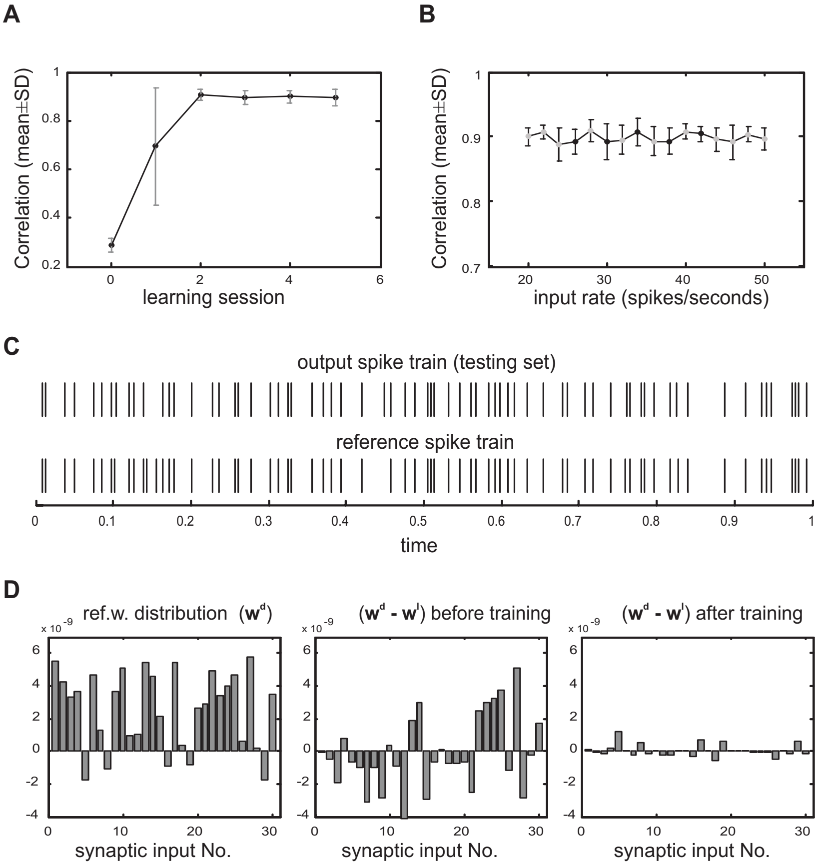

Fig. 1: Function approximation and generalization ability of a single spiking neuron. (A) Correlation of the output and reference spike trains in the consecutive learning sessions. Mean and standard deviation (SD) values of correlation computed for 5 input patterns used during the learning. (B) Generalization ability of the trained neuron is demonstrated by the comparable levels of correlations (mean and SD) computed for the learning (black dots) and validation sets (gray dots). (C) Output spike train of the trained neuron compared to the signal produced by the reference neuron. Both signals generated in response to the same input pattern used only in a validation set. (D) Weight distribution $\bar{w}^d$ of the synaptic inputs of the reference neuron (left). The difference between $\bar{w}^d$ and the corresponding synaptic inputs of the learning neuron $\bar{w}^l$ computed before the training (middle) decreases significantly after the training (right).

4 Discussion

In the presented experiments we observed that the considered learning neuron could approximate the I/O transformations defined by the reference neuron with

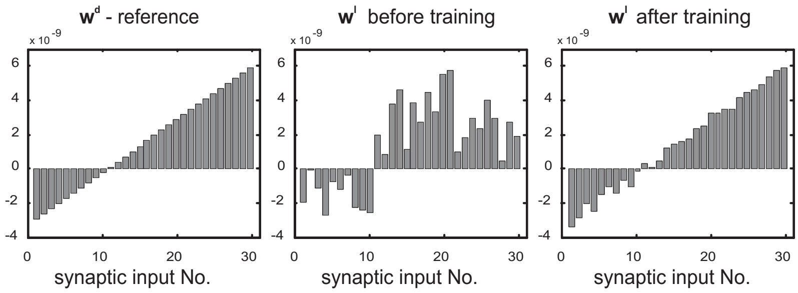

Fig. 2: Another variant of the generalization experiment with randomly generated input spike patterns and (A) the uniform distribution of weights in a reference neuron $\bar{w}^d$. (B) Initial weight distribution $\bar{w}^l$ of the synaptic inputs in n^l. (C) Weight distribution $\bar{w}^l$ after 5 learning sessions.

a satisfactory precision and that the learning process converged quickly.

On the other hand, the experiments demonstrated that during learning the distribution $\bar{w}^l$ reached some stable state (denote $\bar{w}^*$), similar, but not identical to $\bar{w}^d$. This suggests that $\bar{w}^*$ is a local attractor (in the weight space) of the learning process under the given conditions of the experiments and for the given target spike patterns. This demonstrates that many local attractors are possible, $\bar{w}^*$ represents one of them and $\bar{w}^d$ represents another one (according to ReSuMe rules, see eq.(1)). The initial weight distribution of the learning neuron determines which attractor is finally reached.

The existence of many local attractors can be explained by realizing that the definition of some target spike train $S^d(t)$ expected at the neuron's output does not uniquely determine the time course of the membrane potential $V_m(t)$ in that neuron. On the other hand, the equations of neuron's dynamics uniquely define the relation between $V_m(t)$ and the corresponding input weights. Thus there are potentially many weight distributions $\bar{w}^*$ "generating" the same spike pattern $S^d(t)$.

However, longer $S^d(t)$ patterns define more constraints for the possible $V_m(t)$ and thus determine these traces more precisely. In that way also the number of the corresponding distributions $\bar{w}^*$ is reduced. This effect was observed in our simulations (not illustrated here). We hypothesize that the number of attractors can be reduced to a single, global attractor $\bar{w}^* = \bar{w}^d$ only for the very long $S^d(t)$ patterns and in the worst case for the patterns of infinite length.

5 Conclusions

In this paper we demonstrated that SNN trained with ReSuMe can effectively learn a wide class of input to output transformations and that the trained neurons have the expected generalization property. The positive results point out to the suitability of the ReSuMe method as a bearing approach for the real-life applications in which the spiking neural networks could be effectively trained to perform the tasks of approximation, classification or control.

Our results are similar to those reported in [7] where authors trained a spiking neuron to reproduce the predefined I/O transformations by applying the Supervised-STDP method. However, that method has a significant disadvantage resulting from the STDP properties. Namely, STDP always produces bimodal distribution of weights, where each weight evolves toward its minimal or maximal possible value. For this reason it is problematic to obtain intermediate stable weight distribution during the training process. This problem requires some special modifications of the learning rules [7]. In contrast, ReSuMe approach is able to produce stable solution for any arbitrary chosen distribution of weights, which was demonstrated in our experiments. We also demonstrated the positive learning results with the neurons driven by both excitatory and inhibitory synapses.

In our recent work we trained only a single spiking neuron. The natural direction of the future works is to train more complex structures of spiking neurons. It would be also interesting to test our approach on the stochastic neurons and synapses.

References

[1] John Hertz, Anders Krogh, and Richard Palmer. *Introduction to the Theory of Neural Networks*. Addison-Wesley, 1991.

[2] Filip Ponulak. ReSuMe - new supervised learning method for Spiking Neural Networks. Technical Report, Institute of Control and Information Engineering, Poznan University of Technology, 2005. Available at `http://d1.cie.put.poznan.pl/~fp/`.

[3] Wulfram Gerstner and Werner Kistler. *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, Cambridge, 2002.

[4] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[5] Werner Kistler. Spike-timing dependent synaptic plasticity: a phenomenological framework. *Biol.Cybern.*, 87:416–427, 2002.

[6] Jean-Pascal Pfister, Taro Toyoizumi, David Barber, and Wulfram Gerstner. Optimal Spike-Timing Dependent Plasticity for Precise Action Potential Firing. arXiv:q-bio. NC/0502037. v1. 24 Febr. 2005, available at: `http://diwww.epfl.ch/~jpfister/papers/Pfister_05a.pdf`.

[7] Robert Legenstein, Christian Naeger, and Wolfgang Maass. What can a Neuron Learn with Spike-Timing-Dependent Plasticity?, 2005. Submitted for publication.

[8] Andrzej Kasiński and Filip Ponulak. Experimental Demonstration of Learning Properties of a New Supervised Learning Method for the Spiking Neural Networks. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Biological Inspirations*, volume 3696 of *Lecture Notes in Computer Science*, pages 145–153, 2005.

[9] The IGI LSM Group. CSIM: A Neural Circuit SIMulator. Technical University, Graz, `http://www.lsm.tugraz.at`.