A novel biologically plausible supervised learning method for spiking neurons

Lilin Guo, Zhenzhong Wang, and Malek Adjouadi

Center for Advanced Technology and Education Department of Electrical and Computer Engineering College of Engineering and Computing Florida International University, Miami, Florida, USA

Abstract – A novel learning rule, Cross-Correlated Delay Shift (CCDS) learning algorithm, is proposed for processing spatiotemporal patterns in this study. CCDS is a supervised learning rule that is able to learn association of arbitrary spike trains in a supervised fashion. Single spiking neuron trained according to CCDS algorithm is capable of learning and precisely reproducing arbitrary target sequences of spikes. Unlike the ReSuMe learning rule, synapse delays and axonal delays in CCDS are variants which are modulated together with weights during learning. Besides biological plausibility, CCDS is also computationally efficient. In the presented experimental analysis, the proposed learning algorithm is evaluated by it properties including its robustness in dealing with noisy environment, and its adaptive learning performance to different spatio-temporal patterns. Simulation results have shown that the proposed CCDS learning method achieves learning accuracy and learning speed improvements comparable to ReSuMe.

Keywords: Supervised learning, Spiking neuron, Delay Learning, Spike pattern association

1 Introduction

In recent years, supervised learning in a network of spiking neurons has gained increased attention in diverse machine learning applications. One reason for this interest is that learning from instructions or demonstrations is a fundamental property of our brain to acquire new knowledge and develop new skills. Several supervised learning algorithms have been successfully developed for nonlinear benchmark problems. Some of the existing supervised learning rules, such as SpikeProp [1], QProp [2], RProp [2] etc. are using error back propagation similar to the traditional Neural Network (NN). The two major limitations of these methods and their extensions [2]-[4] are that (1) they do not allow multiple spikes in the output spike train, and (2) are sensitive to spike loss, in that no error gradient is defined when the neuron does not fire for any pattern, and hence will never recover. The tempotron learning rule [5], another gradient descent based approach which is efficient for binary temporal classification task, has these two problems as well. As demonstrated in study [6], non-gradient-based methods

like evolutionary strategies do not suffer from these tuning issues. An evolutionary strategy is, however, time consuming for large-scale networks. Other temporal learning rules, such as SPAN [7], PSD [8], Chronotron [9], have been developed to train neurons to generate multiple output spikes in response to a spatio-temporal stimulus. In the Chronotron, both analytically-derived (E-learning) and heuristically-defined (Ilearning) rules are introduced. Both the E-learning rule and the SPAN rule are based on error function of the difference between the actual output spike train and the desired spike train. Their application is therefore limited to tractable error evaluation, which is unavailable in biological neural networks and is computationally inefficient. The I-learning rule of Chronotron is based on particular case of Spike Response Model, which might have limitations for other spiking neuron models. In addition, it depends on weight initialization. Those synapses with zero initial value will not be updated according to the I-learning rule, which will lead to information loss from afferent neurons.

Well known biologically inspired spike-timing dependent plasticity (STDP) was observed through experiments on hippocampal neurons [10] which directly related the synaptic weight value to the time differential between the pre and post-synaptic firing times. ReSuMe [11] is one of few supervised learning algorithms that based on a learning window concept similar to STDP. Similar to SPAN and PSD, ReSuMe is derived from the Widrow-Hoff rule [12]. It combines STDP and anti-STDP learning window under remote supervision of instruction neuron to produce a desired output spike train in response to a spatiotemporal input spike pattern. With this method, it also can reconstruct the target input/output transformation.

The importance of delays in computing with spiking neurons defining a supervised learning rule acting on the delays of connections (instead of weights) between the reservoir and the readout neurons was well demonstrated in [13]. Axonal conduction delays refer to the time required for an action potential to travel from its initial site near the neuronal soma to the axon terminals, where synapse connect the soma with other neurons. Evidence shows that conduction delay in the mammalian brain can reach from a few ms up to over 50 ms [14]. The effect of delay on the processing ability

of the nervous system has been studied in great detail [15], [16]. There is biological evidence that the synaptic delay can be modulated instead of always being invariant [17]. Such evidence supports the introduction of a novel learning algorithm for spiking neurons. Two known approaches for delay learning in SNNs are delay selection [3], [18] and delay shift [19]. In the delay selection method, two neurons are assumed to be connected by multiple synapses with different fixed delays. The weights of connections when related to suitable delays are enhanced while the weights related to unsuitable ones are decreased. Delay shift method adapts the actual delay values of the connections between neurons during training. Input spike patterns close to the synaptic delay vector will make the neuron emit an output spike. Such adaptation may be achieved by changing the length or thickness of dendrites and axons, the extent of myelination of axons, or the density and type of ion channels [20]. However, the weights in study [19] are considered constant during the learning procedure.

In this paper, a novel supervised learning method, called cross-correlated delay shift (CCDS), is proposed to improve ReSuMe by integrating synaptic delay, axonal delay learning with the synaptic weights learning process.

2 Methods

In this new learning method, the synapse delays and axonal delays associated with weights are obtained in the training phase. The neuron model used in this study is described in section 2.1, ReSuMe method is outlined in section 2.2, and details on the CCDS learning rule are given in section 2.3.

2.1 Spiking neuron model

Simple phenomenon models with low computational cost are more popular for studying the dynamics of spiking neural networks as compared to the more detailed conductance-based neuron model. The 1-D leaky integrate-and-fire model is considered in this study, and the dynamic of the *i*-th neuron is as defined in the following equation:

$$\tau_i \frac{dV_i}{dt} = E - V_i + (I_{syn} + I_{ns}) \cdot R_i \tag{1}$$

where V_i is the membrane potential, $\tau_i = R_i C_i$ is the time constant of membrane, *E* defines the resting potential, I_{syn} and I_{ns} are the synaptic current and background noise current, respectively. Note that when membrane voltage V_i reaches the threshold level V_{th} , the neuron emits a spike and V_i is reset to V_{rest} for a refractory period t_{ref} . The synaptic current is thus modeled as

$$I_{syn}(t) = \sum_{j} w_{j} I_{PSC}^{j}(t)$$
⁽²⁾

where w_j defines the synaptic efficacy of the *j*-th afferent neuron, I_{PSC}^{j} represents the postsynaptic current from afferent spikes. The postsynaptic current with synaptic delay can thus be written as:

$$I_{PSC}^{j}(t-dt_{j}) = \sum_{t^{f}} K(t-t^{m}-dt_{j})H(t-t^{m}-dt_{j})$$
(3)

where t^m and dt_j are the *m*-th spike and the synaptic delay from the *j*-th afferent neuron, respectively; H(t) is the Heaviside function; *K* refers to a normalized exponential kernel function as:

$$K(t) = V_0(\exp(-t/\tau_s) - \exp(-t/\tau_f))$$
(4)

where V_0 is the normalized factor, τ_s and τ_f are the slow and fast decay time constant, respectively, with $\tau_s / \tau_f = 4$.

2.2 ReSuMe

Supervised learning in temporal encoded SNNs attempts to link the input spike train with output spike sequence. ReSuMe is such a learning method which adjusts the synaptic weights of a neuron to generate a desired spike train $S^{d}(t)$ in response to a spatio-temporal input spike pattern $S^{in}(t) = [s_1(t), s_2(t), \dots, s_n(t)].$

In ReSuMe, synaptic weights are modified according to

$$\frac{d}{dt}w(t) = \left[S^d(t) - S^o(t)\right] \left[a + \int_0^\infty W(s)S^{in}(t-s)ds\right]$$
(5)

where $S^{d}(t)$, $S^{in}(t)$ and $S^{o}(t)$ are the desired, pre-and postsynaptic spike trains, respectively. The parameter a is a non-Hebbian term. In the case of excitatory synapses, the term ais positive and the learning window W(s) has a shape similar as in STDP. In the case of inhibitory synapses, a is negative and W(s) is defined similarly as for the anti-STDP rule. When the number of spikes in the actual output spike train $S^{o}(t)$ is more or less than the number of spikes in the desired spike train $S^{d}(t)$, a decrease/increase is assumed in the weights. This will speed up the convergence of the training process. In ReSuMe, no delay was considered.

2.3 CCDS

Taking into consideration both the synaptic and axonal delays, Fig. 1 illustrates a neuron structure with multi-path connectivity. Each spike from the afferent neuron will result in a post-synaptic current (PSC). The membrane potential of the post-synaptic neuron is a weighted sum of all incoming PSCs from afferent neurons. Fig. 1 shows a multi-connected neuron structure with axonal delays d_i , i=1,...,n and

synapse delays dt_i i = 1,...,n. Their corresponding weight values are w_i i = 1,...,n, respectively.



Fig. 1: Neuron structure with multi-path connectivity: axonal delays and synapse delays are from d_1 to d_n and dt_1 to dt_n with corresponding weight values w_1 through w_n , respectively.

The time differential between input and output spike times can be formulated as follows:

$$\delta_{t_i} = t_{post} - (t_{pre} + d_i + dt_i), i = 1, \cdots, n$$
(6)

Then, the positive half of the learning window of spiketiming-dependent plasticity (STDP) results in long-term potentiation (LTP) of the synaptic weights as expressed below:

$$\delta w_i = A_1 \exp(-\frac{\delta t_i}{\tau_1}) \tag{7}$$

where A_1 is the maximum value of the weight potentiation, τ_1 is the width of the window for LTP and δ_{t_i} is the time differential as defined by (6).

Similarly, the negative part of the learning window where long-term depression (LTD) occurs is defined as

$$\delta w_i = A_2 \exp(-\frac{\delta t_i}{\tau_2}) \tag{8}$$

where A_2 is the maximum value of weight depression and τ_2 defines the width of the window for LTD.

The weight modulation can be written as

$$w_{i(new)} = w_{i(old)} + \delta w_i \tag{9}$$

Let us first consider a simple example in order to formulate the relative occurrence rule. Assume both data groups d_1 and d_2 has a total of k spikes occurring at various times within a temporal window T. Consider a particular spike time t_s occurring at n different channels (neurons). These are divided into M groups, g_i , i = 1,...,M, with m channels in each group. t_s occurs p times within group g_1 and q times within group g_2 , Therefore, occurrence of t_s in g_1 relative to g_2 is

$$O(g_1) = \frac{p}{p+q} \tag{10}$$

Similarly, the relative occurrence of t_s in g_2 can be expressed as

$$O(g_2) = \frac{q}{p+q} \tag{11}$$

Then the weight that reflects the association of t_s with g_1 is modified to

$$w_{ij(new)} = w_{ij(old)} + \frac{p}{p+q} \,\delta w_{ij}(t_s) \tag{12}$$

where $w_{ij(old)}$ is the pre-trained value associate with connection w_{ii} .

A similar rule that reflects the association of t_s with g_2 is given by

$$w_{ij(new)} = w_{ij(old)} + \frac{q}{p+q} \delta w_{ij}(t_s)$$
(13)

Dividing all input spike trains into M groups, each group having m spike trains, the updated weight can be written as

$$w_{ij(new)} = w_{ij(old)} + C_c \cdot \delta w_{ij}(t_s) \tag{14}$$

where the cross correlated term is given by the relation

$$C_{c} = \frac{\sum_{N_{i}^{f}=0}^{N_{i}^{f}=k} O(g_{i})}{\sum_{N_{i}^{f}=0}^{N_{i}^{f}=k} \sum_{j=1}^{m} O(g_{j})}$$
(15)

The proposed CCDS algorithm is a heuristic method which helps the neuron generate a desired output with the ability of removing undesired output instances. In CCDS, the delay is applied to the connection that has the nearest spike before the desired time, which leads to an increase in postsynaptic potential (PSP) at the desired time. In addition, the reduction of the PSP for undesired output spike is achieved by delayed PSP. The reduction may eventually cancel undesired spikes.

The nearest previous input spike is calculated via local variable, $x_i(t)$, described in (16) below:

$$x_{i}(t) = \begin{cases} A_{o} \exp(-(t - t_{i}^{f}) / \tau), & \text{for } t_{i}^{f} < t < t_{i}^{f+1} \\ A_{o} & \text{for } t = \cdots t_{i}^{f}, \ t_{i}^{f+1}, \cdots \end{cases}$$
(16)

where amplitude A_o and time decay τ are constants. $x_i(t)$ in this case jumps to a saturated value A_o whenever a presynaptic spike arrives.

If the previous spike is far from the current time t, then $x_i(t)$ is low, otherwise if it is close to t, then $x_i(t)$ is high. The delays d_i and dt_i shift the effect of its spike to time t by using the inverse operation of (16) as expressed below:

$$d_i + dt_i = t - t_o^f = -\tau \ln\left(\frac{x_o(t)}{A_o}\right) \tag{17}$$

At desired spiking time without any actual output spikes, $x_o(t)$ is chosen from excitatory synapses that are not delayed previously. The chosen connection is delayed by $d_i + dt_i$. Then the spike is shifted toward the desired time, which will lead into an increment in the PSP. In contract, at the undesired spiking time with output spikes, $x_o(t)$ is chosen from inhibitory synapses that are not delayed previously.

Considering both the cross-correlation and delay shift effect, the weights as governed by the CCDS learning rule are updated on the basis of (18).

$$\frac{dw_{i}(t)}{dt} = C_{c}[s_{d}(t) - s_{o}(t)] \cdot [a + \int_{0}^{+\infty} W(s - d_{i} - dt_{i})S_{i}(t - d_{i} - dt_{i} - s)ds]$$
(18)

where C_c is the same as in (12), with the learning window being

$$W(s) = \begin{cases} Ae^{-\frac{s}{\tau}}, s \ge 0\\ 0, s < 0 \end{cases}$$
(19)

3 Results

3.1 Experimental setup

The trained neuron is connected with *n* afferent neurons, and each fires a spike train in the time interval (0, T). Input spike trains are desired spike train and are randomly generated with a homogeneous Poisson distribution with mean frequency F_{in} and F_d , respectively. The ratio of inhibitory and excitatory synapses is set to the standard ratio of 1/4 as cortical neuron [21]. The initial synaptic weights are drawn randomly from uniform distribution with mean value of -0.5 and a standard deviation of 0.2 for inhibitory synapse, and with mean value of 0.75 and a standard deviation of 0.2 for excitatory one. For the learning parameters, we set the membrane time constant $\tau_i = 10ms$; the refractory period $t_{ref} = 5ms$; the initial voltage, the threshold voltage and the reset voltage are selected as $V_{init} = -60mV$, $V_{th} = -55mV$ and $V_{reset} = -65mV$, group number M=20, number each channels m=30, respectively. The weights are capped in the range [-15, 15] to ensure convergence. At the beginning of the training phase, 20% of the weights are considered inhibitory while 80% of the weights are considered excitatory. In each epoch, synaptic delay and axonal delay are assumed to be adjusted only once. In contrast, the connection weight can be changed many times during the learning phase.

As axonal delays and synapses delays are limited in the biological neurons, all axonal delays and synaptic delays in this method evolve within the interval [0, 40]ms and [0, 2]ms, respectively.

3.2 Learning process

The correlated-based metric(C) [22] is used to evaluate the similarity of the desired spike pattern with the actual output spike train. It takes values between zero and one. The metric C equals one for identical spikes and drops to zero for loosely correlated trains.



Fig. 2: Training results without noise. (a) V_m : membrane potential after learning; red dots: target spike train; green dots: actual output spike train; (b) correlated-based metric C of target and output spike trains.

Input spike trains are generated by a homogenous Poisson spike train with frequency $F_i = 10Hz$ with *n* afferent neurons (*n*=600). Frequency $F_d = 40Hz$ is chosen to produce the output spike train. Delayed version LIF is utilized for the training. In Fig. 2(a), the red dots are the target spikes while the green dots are the actual spikes. In Fig. 2(b), at around 16 epochs, the correlation C of desired and observed output spike trains reach a satisfactory level C>0.95. After a small period oscillating, the correlation C converges towards 1. The evolution of firing patterns generated by the neuron in consecutive learning epochs can be seen in Fig. 3(c), where the cyan line is the desired spike and the blue dots are the actual output spike patterns according to the learning epochs. Fig. 3(a) and Fig. 3(b) present the membrane voltage of learned neuron before learning and after learning, respectively.

The results show that the neuron can successfully learn to emit the desired spike train from the initial random output spike train after just 69 learning epochs. The six randomly generated spike patterns converge perfectly after training.



Fig. 3: Temporal sequence learning of a typical run without noise (a) membrane potential before learning; (b) membrane potential after learning; (c) learning process.

3.3 Adaptive learning performance

At the beginning, the neuron is trained to learn a target train as in the previous experiments. After successfully learning the process, the target spike train is changed to an arbitrarily generated train, where the precise spike time and firing rate may be different from the previous target train. We found that, we could successfully train the neuron to learn the new target within several epochs with the CCDS learning rule. As shown in Fig. 4(a), each dot denotes a spike. At the beginning, the neuron is trained to learn one target (denoted by cyan bar in the bottom part). After 100 epochs of learning (the dashed blue line), the target is changed to another randomly generated train (denoted by the cyan bar in the top part). Again, the neuron successfully learned the new target spike train within 60 epochs. Fig. 4(b) shows the correlated measure C of different desired spike train and output spike train along the learning process.



Fig. 4: Adaptive learning of different target trains (a) sequence learning with the changed target train; (b) correlated-based metric *C* of target and output spike trains.

3.4 Robustness to noise

In the previous experiments, the simple case where the neuron is trained to learn a single pattern without noise is assumed. However, in practical settings, the reliability of the results could be significantly affected by the presence of noise. ReSuMe is shown to be robust to noise during the learning process [11]. Here, we re-evaluate the robustness of the proposed CCDS learning rule.



Fig. 5: Temporal sequence learning of a typical run with noise

In this experiment, a LIF neuron with n=600 afferent neurons under background current noise is tested. Gaussian noise is added to the LIF neuron where $I_{ns} = 0.2$ nA. Randomly generated Poisson spike trains are used for both the input and desired spike trains. As shown in Fig. 5, eight spike patterns still converge within 50 epochs. Even when more patterns are considered in the presence of noise, the results still converged within 50 epochs as illustrated in Fig. 5 and Fig. 6.



Fig. 6: Synaptic weights during CCDS supervised learning with noise *Ins*=0.2nA

3.5 Comparison with ReSuMe

In the following experiments, each spike train has a total time duration of T = 400ms. At the beginning of CCDS simulation, none of the input spike trains have delays. The same input spike trains with $F_{in} = 5Hz$ and desired spike train $F_d = 100Hz$ are selected for both CCDS and ReSuMe. The performance of the proposed method is compared with that of ReSuMe in Fig. 7. One can note that CCDS learning rule achieves high learning accuracy much faster than ReSuMe. The evolution of weights for each method is given in Fig. 8(a) and Fig. 8(b), respectively. The CCDS method managed to reach the satisfied level C>0.95 much earlier at the 8th epoch and settles on a stable set of weights thereafter. In contract, the ReSuMe training shows the weights continue to adapt even after the 100th epoch.



Fig. 7: Evolution of correlated-based metric C for ReSuMe and CCDS



Fig 8: Evolution of the weights during learning using (a) CCDS rule; (b) ReSuMe rule

4 Conclusions

In this study, the spatio-temporal associations of key events or patterns were investigated using the proposed CCDS training algorithm. By making use of the biological concepts of spike-timing dependent plasticity (STDP), axonal delays, and synapse delays, the CCDS is able to learn the association between precise test patterns. The results obtained confirm that the proposed method is highly effective and computationally efficient in the spatio-temporal association of arbitrary spike trains in a supervised fashion. Future work will focus on application of proposed learning method to realworld problems such as detecting interictal spikes in electroencephalography (EEG) data and extend the single neuron learning algorithm to network level to do the spatiotemporal pattern classification.

5 Acknowledgements

This research is supported through NSF grants CNS-0959985, CNS-1042341, HRD-0833093, IIP 1338922 and

IIP-1230661. The support of the Ware Foundation is greatly appreciated.

6 References

- Sander M. Bohte, Joost N. Kok, and H. La Poutre, "Error-backpropagation in Temporally Encoded Networks of Spiking Neurons," *Neurocomputing*, vol. 48, pp. 17–37, 2002.
- [2] S. McKennoch, Dingding Liu, and L. G. Bushnell, "Fast Modifications of the SpikeProp Algorithm," *IJCNN '06. Int. Jt. Conf. Neural Networks, 2006.*, pp. 3970–3977, 2006.
- [3] S. Ghosh-Dastidar and H. Adeli, "Improved Spiking Neural Networks for EEG Classification and Epilepsy and Seizure Detection," *Integr. Comput. Aided. Eng.*, vol. 14, no. 3, pp. 187–212, 2007.
- [4] S. B. Shrestha and Q. Song, "Adaptive learning rate of SpikeProp based on weight convergence analysis," *Neural Networks*, vol. 63, pp. 185–198, Dec. 2015.
- [5] R. Gutig and H. Sompolinsky, "The Tempotron: a Neuron that Learns Spike Timing-Based Decisions," *Nat. Neurosicence*, vol. 9, pp. 420–428, 2006.
- [6] A. Beltreche, L. P. Maguire, M. Mcginnity, and Q. Wu, "Evolutionary Design of Spiking Neural Networks," *New Math. Nat. Comput.*, vol. 02, no. 03, pp. 237–253, Nov. 2006.
- [7] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, "SPAN: Spike Pattern Association Neuron for Learning Spatio-Temporal Spike Patterns," *Int. J. Neural Syst.*, vol. 22, no. 4, p. -1, 2012.
- [8] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Precise-Spike-Driven Synaptic Plasticity: Learning Hetero-Association of Spatiotemporal Spike Patterns," *PLoS One*, vol. 8, no. 11, p. e78318, 2013.
- [9] R. V Florian, "The Chronotron: A Neuron That Learns to Fire Temporally Precise Spike Patterns," *PLoS One*, vol. 7, no. 8, pp. 1–27, 2012.
- [10] R. P. N. Rao and T. J. Sejnowski, "Spike-Timing-Dependent Hebbian Plasticity as Temporal Difference Learning," *Neural Comput.*, vol. 13, no. 10, pp. 2221– 2237, 2001.
- [11] F. Ponulak and A. Kasinski, "Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification, and Spike Shifting," *Neural Comput.*, vol. 22, pp. 467–510, 2010.

- [12] B. Widrow and M. Hoff, "Adaptive switching circuits.," *1960 IRE WESCON Conv. Rec.*, no. 4, pp. 96 – 104, 1960.
- [13] A. G. Ioana Sporea, "Supervised Learning in Multilayer Spiking Neural Networks," *Neural Evol. Comput.*, vol. 25, no. 2, pp. 473–509, 2013.
- [14] H. A. Swadlow, "Physiological properties of individual cerebral axons studied in vivo for as long as one year," J. Neurophysiol., pp. 1346–1362, 1985.
- [15] B. Glackin, J. A. Wall, T. M. McGinnity, L. P. Maguire, and L. J. McDaid, "A spiking neural network model of the medial superior olive using spike timing dependent plasticity for sound localization.," *Front. Comput. Neurosci.*, vol. 4, 2010.
- [16] M. Gilson, M. Bürck, A. N. Burkitt, and J. L. van Hemmen, "Frequency Selectivity Emerging from Spike-Timing-Dependent Plasticity," *Neural Comput.*, vol. 24, no. 9, pp. 2251–2279, 2012.
- [17] J. W. Lin and D. S. Faber, "Modulation of synaptic delay during synaptic plasticity," *Trends Neurosci.*, vol. 25, no. 9, pp. 449–455, 2002.
- [18] S. Ghosh-Dastidar and H. Adeli, "A New Supervised Learning Algorithm for Multiple Spiking Neural Networks with Application in Epilepsy and Seizure Detection," *Neural Networks*, vol. 22, no. 10, pp. 1419–1431, 2009.
- [19] P. Adibi, M. R. Meybodi, and R. Safabakhsh, "Unsupervised learning of synaptic delays based on learning automata in an RBF-like network of spiking neurons for data clustering," *Neurocomputing*, vol. 64, no. 1–4 SPEC. ISS., pp. 335–357, 2005.
- [20] R. Wang, G. Cohen, K. M. Stiefel, T. J. Hamilton, J. Tapson, and A. van Schaik, "An FPGA Implementation of a Polychronous Spiking Neural Network with Delay Adaptation," *Front. Neurosci.*, vol. 7, no. February, pp. 1–14, 2013.
- [21] J. Mishra, J. M. Fellous, and T. J. Sejnowski, "Selective attention through phase relationship of excitatory and inhibitory input synchrony in a model cortical neuron," *Neural Networks*, vol. 19, no. 9, pp. 1329–1346, 2006.
- [22] S. Schreiber, J. M. Fellous, D. Whitmer, P. Tiesinga, and T. J. Sejnowski, "A new correlation-based measure of spike timing reliability," *Neurocomputing*, vol. 52–54, pp. 925–931, 2003.