2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

A New Learning Rule for Classification of Spatiotemporal Spike Patterns

Qiang Yu
Department of Electrical and Computer Engineering
National University of Singapore
Singapore 117576
Email: yu.qiang@nus.edu.sg

Huajin Tang
Institute for Infocomm Research
Agency for Science Technology and Research (A*STAR)
Singapore 138632
Email: htang@i2r.a-star.edu.sg

K.C. Tan
Department of Electrical and Computer Engineering
National University of Singapore
Singapore 117576
Email: eletankc@nus.edu.sg

***Abstract*—In this paper, we present a new learning rule for classification of spatiotemporal spike patterns. This rule is derived from the common Widrow-Hoff rule, and it can be used for both the association and the classification. We mainly focus on investigating its classification ability in this paper. Through experimental simulations, it can be seen that this rule can successfully train the neuron to reproduce the desired spikes. In the classification task, the neuron is capable to classify different categories with the learning rule. We have proposed two decision-making schemes which are the absolute confidence and the relative confidence criteria. The classification performance is largely improved by the relative confidence criterion. The performance of this rule on classification of spatiotemporal spike patterns is also investigated and benchmarked by the tempotron rule.**

I. Introduction

The great computational power of human brain has attracted numerous researchers into the area of computational neurobiology during these several decades. How the information is represented in the brain still remains unclear. However, there is strong evidence showing that the representation of external stimuli in the brain is in a form of spatiotemporal spikes [1], [2]. Neurons in the brain propagate information through action potentials (or called as spikes). These discoveries suggest the development of the spiking neural networks (SNNs). SNNs are more biologically plausible and computational powerful than the traditional artificial neural networks.

The coding schemes consider how the spatiotemporal spikes convey the information of external stimuli. Among different coding schemes, rate coding and temporal coding [3] are two of the most widely studied methods. The rate coding is the most basic example of a neural code in which information is conveyed through spike count within a time window. However, the precise timing of each spike is considered in the temporal coding. Recently, experimental evidence suggests that neural systems also use exact time of spikes to convey information. Neurons are revealed to precisely respond to stimuli on a millisecond timescale in the retina [4], [5], the lateral geniculate nucleus (LGN) [6] and the visual cortex [7], [8]. These observations support the hypothesis of the temporal coding. Additionally, recent studies have shown that the temporal coding scheme can carry more information than the rate coding scheme [9]–[11].

Several learning algorithms have been proposed for processing spatiotemporal spike patterns. According to different learning algorithms, the neurons adapt their synaptic weights during learning and store these synaptic weights as memories after learning. One of the first supervised learning algorithms for SNN is SpikeProb [12]. SpikeProb is a gradient descent based learning rule. This rule can solve nonlinear classification tasks by emitting single spikes at desired firing time. However, SpikeProb in its original form cannot learn to reproduce multiple spike train. The tempotron rule [13], which also uses a gradient descent approach, is evaluated to be efficient in distinguishing binary temporal classification task. The tempotron makes the decision by a binary response of firing or not. Since the tempotron is designed for recognition, it is also unable to produce precise spikes. Several learning algorithms, such as ReSuMe [14], [15], Chronotron [16], SPAN [17] and Precise-Spike-Driven (PSD) synaptic rule [18], have been proposed to train neurons to precisely respond to spatiotemporal spike patterns. Among these four rules, without complex error calculation, the PSD rule is simple and efficient in the view point of calculation, and yet biologically plausible [18]. The PSD rule is derived from the Widrow-Hoff (WH) rule by applying the spike convolution method on the afferent spike trains. The synaptic adaptation in the PSD rule is driven by the error between the desired and the actual output spikes, with positive errors causing long-term potentiation (LTP) and negative errors causing long-term depression (LTD). The amount of the adaptation depends on the eligibility trace determined by the afferent spikes. The PSD rule can perform both the association task and the classification task.

In this paper, we further investigate the ability of the PSD rule on classification of spatiotemporal spike patterns. In the following, the detailed description of the PSD rule is presented. Through simulation, we demonstrate several preliminary results of the learning rule. In the first experiment, the association ability of the PSD rule is demonstrated. With the PSD rule, the neuron could be trained to successfully reproduce desired spikes. In the following experiments, the PSD rule is applied to perform the classification task of

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

spatiotemporal spike patterns. Two decision making criteria, namely the absolute confidence and the relative confidence, are proposed for recognition. The performance of the PSD rule for classification is also analyzed and discussed.

II. Learning Rule

In this section, we firstly describe the neuron model used in this study. The PSD rule is presented later through detailed description.

A. The neuron model

Spiking neuron models have attracted more attention recently because of their biological realism. There are several kinds of spiking neuron models such as the integrate-and-fire (IF) model [19], the resonate-and-fire model [20], the Hodgkin-Huxley model [21], and the Izhikevich model (IM) [22].

The IF model is the most widely used spiking neuron model because of its simplicity and computational efficiency. For the sake of simplicity, the leaky IF model is considered here. The neuron evolves according to:

$$\tau_m \frac{dV_m}{dt} = -(V_m - E) + R_m(I_{ns} + I_{syn}) \tag{1}$$

where V_m is the membrane potential, $\tau_m = R_m C_m$ is the membrane time constant, $R_m = 1M\Omega$ and $C_m = 10nF$ are the membrane resistance and capacitance, respectively, E is the resting potential, I_{ns} and I_{syn} are the background noisy current and synaptic current, respectively. When V_m exceeds a constant threshold V_{thr}, the neuron is said to fire, and V_m is reset to V_{reset} for a refractory period t_{ref}. We set $E = V_{reset} = 0mV$, $V_{thr} = E + 18mV$ for clarity, but any other values as $E = -60mV$ will result in equivalent dynamics as long as the relationships among E, V_{reset} and V_{thr} are kept.

For the postsynaptic neuron, the input synaptic current is described as:

$$I_{syn}(t) = \sum_i w_i I^i_{PSC}(t) \tag{2}$$

where w_i is the synaptic weight of the i-th afferent neuron, and I^i_{PSC} is the un-weighted postsynaptic current from the corresponding afferent.

$$I^i_{PSC}(t) = \sum_{t^j} K(t - t^j)H(t - t^j) \tag{3}$$

where t^j is the timing of j-th spike emitted from the i-th afferent, $H(t)$ refers to the Heaviside function, K denotes the normalized kernel and we choose as:

$$K(t - t^j) = V_0(\exp(\frac{-(t - t^j)}{\tau_s}) - \exp(\frac{-(t - t^j)}{\tau_f})) \tag{4}$$

where V_0 is a normalization factor so that the maximum value of the kernel is 1, τ_s and τ_f are the slow and fast decay constants, respectively. Their ratio is fixed at $\tau_s/\tau_f = 4$.

Fig. 1 illustrates the neuron structure. Each spike from the afferent neuron will result in a postsynaptic current (PSC). The membrane potential of the postsynaptic neuron is the weighted sum of all incoming PSCs over all afferent neurons.

B. The learning rule

In this part, the PSD rule is presented. This rule is derived from the common Widrow-Hoff (WH) rule (also called as Delta rule). This WH rule is described as:

$$\Delta w_i = \lambda x_i(y_d - y_o) \tag{5}$$

where λ is positive constant referring to the learning rate, x_i, y_d and y_o refer to the input, desired output and actual output, respectively.

Since this WH rule was introduced for traditional neuron models such as perceptron, the variables in WH rule are regarded as real-valued vectors. However, the input and output signals of spiking neurons are represented by the timing of spikes. Direct implementation of this WH rule to spiking neurons is problematic.

A spike train is defined as a sequence of impulses triggered by the particular neuron at its firing times. It is described as $s(t) = \Sigma_f \delta(t - t^f)$. Here, t^f is the f-th firing time, and $\delta(x)$ is the Dirac's function ($\delta(x) = 1$, if $x = 0$; or 0, otherwise).

The products of Dirac functions are mathematically problematic. By applying spike convolution on the input spike train with a kernel as Eq. (4), I_{PSC} can be used as an eligibility trace for weight adaptation. The learning rule becomes:

$$\frac{dw_i(t)}{dt} = \lambda[s_d(t) - s_i(t)]I^i_{PSC}(t) \tag{6}$$

The above equation formulates an online learning rule. By integrating Eq. (6), we get:

$$\begin{aligned} \Delta w_i =& \lambda \int_0^\infty [s_d(t) - s_i(t)]I^i_{PSC}(t)dt \\ =& \lambda\Big[\sum_g \sum_f K(t^g_d - t^f_i)H(t^g_d - t^f_i) \\ &+ \sum_h \sum_f K(t^h_o - t^f_i)H(t^h_o - t^f_i)\Big] \end{aligned} \tag{7}$$

This equation could be used for trial learning where the weight modification is performed at the end of pattern presentation.

Additionally, to measure the distance between two spike trains, we use the van Rossum metric [23] but with a different filter function described as in Eq. (4). This filter is used to compensate the discontinuity of the original filter function. The distance is described as:

$$Dist = \frac{1}{\tau} \int_0^\infty [f(t) - g(t)]^2 dt \tag{8}$$

where τ is a free parameter (we set $\tau = 10\ ms$ here), $f(t)$ and $g(t)$ are filtered signals of two considered spike trains. This distance measurement is not involved in our learning rule, but is used for analyzing the performance.

III. Experimental Results

In this section, simulation experiments are conducted to demonstrate the performance of the PSD rule on classification. The first experiment simply demonstrates the association ability of the PSD rule. The PSD rule can successfully train a

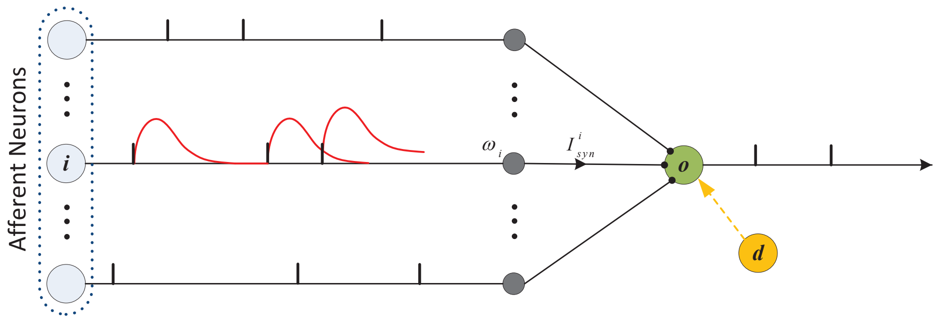

Fig. 1. Illustration of the neuron structure. The afferent neurons are connected to the postsynaptic neuron through synapses. Each emitted spike from the afferent neuron will produce a postsynaptic current (PSC). The membrane potential of the postsynaptic neuron is the weighted sum of all incoming PSCs from all afferent neurons. The yellow neuron denotes the instructor which is used for learning.

neuron to associate the input spatiotemporal spike pattern with the desired spike train. In the latter experiments, a spatiotemporal pattern classification problem is mainly considered and investigated.

A. Demonstration of the learning rule

In this experiment, the neuron is trained to associate a randomly generated spatiotemporal pattern with a specific target spike train. For the sake of simplicity, we only consider a single spike for each afferent neuron. The neuron is connected with n_{pre} afferent neurons, and each fires a single spike in the time interval of $(0, T)$. Each spike is randomly generated with a uniform distribution. We set $n_{pre} = 500$, $T = 200\ ms$ here. To avoid single synapse dominating the firing of the neuron, we limit the weight below $w_{max} = 6\ nA$. The initial synaptic weights are drawn randomly from a normal distribution with mean value of $0.5\ nA$ and standard deviation of $0.2\ nA$. For the learning parameters, we set $\lambda = 0.01 w_{max}$ and $\tau_s = 10\ ms$. The target spike train could be randomly generated, but for simplicity, we specify it as $[40,\ 80,\ 120,\ 160]\ ms$ for this experiment.

Fig. 2 illustrates a typical run of the association ability of the PSD rule. Initially, the neuron seems to fire at arbitrary times and with a different firing rate from the target train, which results in a large spike distance value. The actual output spike train is quite different from the target spike train at the beginning. Along the learning process, the neuron gradually produces spikes at the target times, which is also reflected by the reducing spike distance. After finishing the first 10 epochs of learning, both the firing rate and precise timings meet those in the target spike train. The dynamic of neuron's membrane potential is also illustrated. Whenever the membrane potential exceeds the threshold value, a spike is emitted and then the potential is kept at the reset level for a refractory period. Detailed mathematical description is presented previously.

This experiment shows the feasibility of the PSD rule to train the neuron to reproduce the desired spike train. After several learning epochs, the neuron can successfully spike at the target times. In other words, the PSD rule is able to train the neuron to associate the input spatiotemporal spike pattern with the desired output spike train within several training epochs. The knowledge of the input pattern is stored by a specified spike train. Both of the dimension and the complexity of the input information are largely reduced by this association.

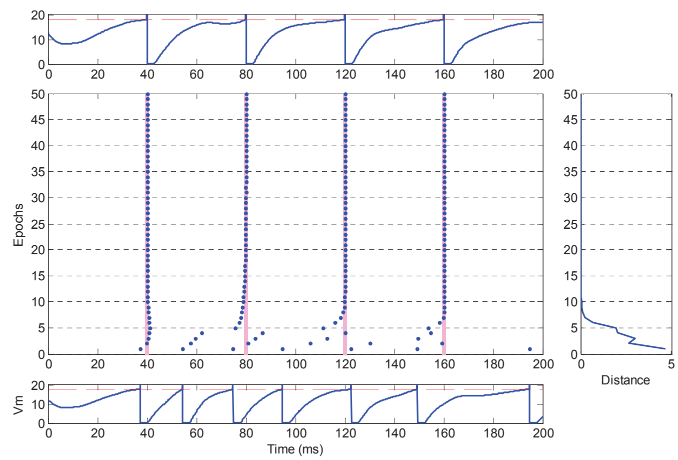

Fig. 2. Illustration of the association ability of the PSD rule in a typical run. The neuron is connected with $n_{pre} = 500$ synapses, and is trained to reproduce spikes at the target times (denoted as the shaded bars in middle). The bottom and top show the dynamics of the neuron's potential before and after learning, respectively. The dashed red lines denote the firing threshold. In the middle, each spike is denoted as a dot. The right figure shows the distance between the actual output spike train and the target spike train.

B. Classification of spatiotemporal patterns

In this experiment, the ability of the PSD rule for classifying spatiotemporal spike patterns is investigated. Multiple classification task is considered. In this experiment, 5 categories are used. Five random spike patterns are generated in a similar fashion as for previous experiment, and they are fixed as the templates. A Gaussian jitter with a standard deviation is used to generate jittered patterns. Fig. 3 illustrates some examples of spike patterns used in this experiment.

1) Decision-making criteria: In this scenario, we investigate the effects of different decision-making criteria on the

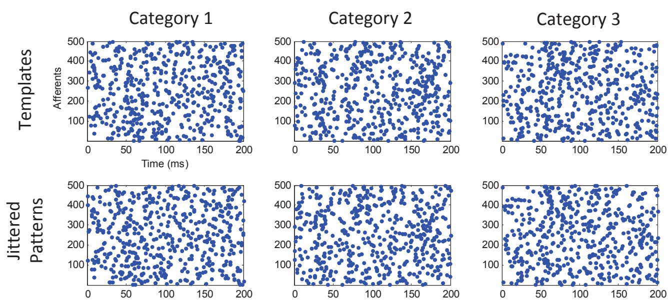

Fig. 3. Examples of spike patterns generated for training and testing. Only three categories are demonstrated here. The top row shows the fixed templates of three different categories. The bottom row shows the patterns generated for training and testing. They are generated by adding a Gaussian time jitter of zero mean and 3 ms standard deviation to the templates. Each dot in the figure denotes a spike.

classification performance. We use a jitter noise of 3 ms to generate both the training set and the testing set. The training set and the testing set contains 5×20 and 5×50 samples, respectively. Five neurons are trained to recognize these five categories, and each neuron corresponds to one category. The training set is used for training and the testing set is for determining the generalization ability of the trained neurons. Different neurons for each category could be specified to fire different spike trains. However, for simplicity, all the neurons in this experiment are trained to fire a same spike train ($[40,\ 80,\ 120,\ 160]\ ms$). The experiment is repeated by 20 runs, and for each run a different initial condition is chosen.

After training, the classification is performed on both the training and the testing set. For the classification task, we propose two decision-making criteria which are the absolute confidence and the relative confidence. Under the absolute confidence criterion, only if the distance between the desired spike train and the actual output spike train of the corresponding neuron is smaller than a specified value (0.5 is used here), the input pattern is regarded as being correctly recognized. For the relative confidence criterion, a scheme of competition is used. The incoming pattern will be labeled by the winning neuron which reproduces the closest spike train to the desired train.

Fig. 4 shows the average classification accuracy for each category under the two proposed decision-making criteria. From the absolute confidence, we see that the neuron could successfully classify the training set with an average accuracy of 99.20% across all categories. The average accuracy for the testing set is 66.74% across all categories. Noteworthily, under the relative confidence, both the average accuracies for the training and the testing set reach to 100%. The classification performance is largely improved by the decision-making criterion of relative confidence. In the absolute confidence, the trained neuron focuses more to exactly recognize those memorized patterns. However, in the relative confidence, the trained neuron focuses more to decide the most possible category through competition. Thus, the relative confidence

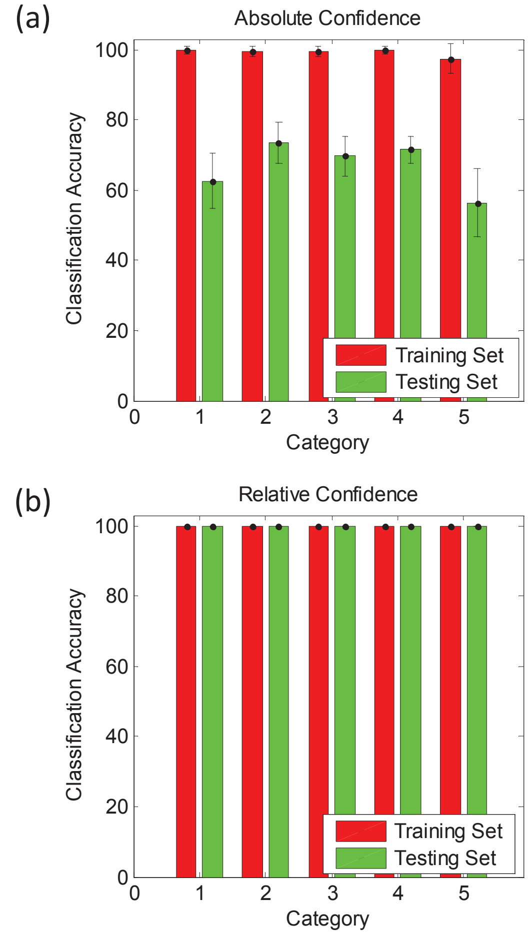

Fig. 4. The average accuracies for the classification of spatiotemporal patterns with different decision-making criteria. There are 5 categories to be classified. The average accuracies are represented by shaded bars. Two types of criteria for making decision are proposed and investigated. (a) is the absolute confidence criterion, and (b) is the relative confidence criterion. All the data are averaged over 20 runs.

criterion can benefit the classification task, and we use this criterion in the following experiment.

2) Effects of the number of desired spikes: In this scenario, we investigate the effects of the number of desired spikes on the classification performance. The desired spike train is set to be evenly distributed over the time window T with a specified number of spikes n. The firing time of the i-th desired spike: $t_i = i/(n+1) \cdot T$, $i = 1,\ 2...n$. Additionally, we also use the tempotron rule [13], [24], under the same experimental setup with the PSD rule, to provide comparisons for analysis.

Jitter noises are added to generate noisy patterns. In the training phase, the learning neurons are trained for 100 epochs with a jitter strength of 2 ms. In each learning epoch, a training set of 50 patterns, with 10 for each category, is generated. After training, a jitter range of 0-14 ms is used to investigate the generalization ability. The number of the testing patterns for each jitter strength is set to 100. All the results are averaged over 20 runs. The effect of n on the classification performance is shown in Fig. 5.

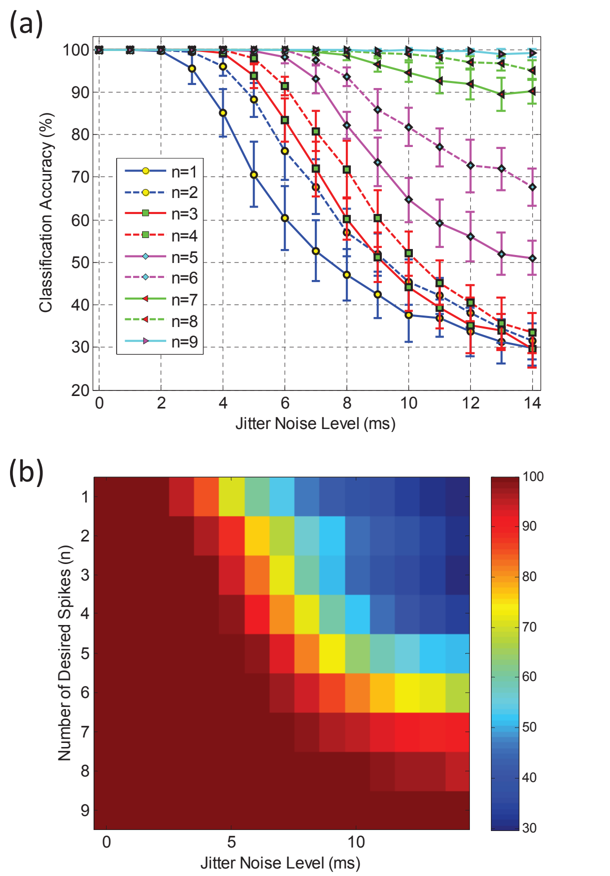

Fig. 5. The performance of the PSD rule with different numbers of target spikes on the spatiotemporal classification task. (a) shows the whole results with both the mean values and the standard deviation values. (b) is another demonstration of the results where the average accuracies are represented by different colors.

As can be seen in Fig. 5, the testing accuracy stays on a high level when the noise is under the training noise. The performance will gradually decrease with an increasing noise strength. When n is low, the classification is also relative low. An increasing number of the desired spikes can improve the classification performance significantly. For example, when $n = 9$, the classification accuracy is still quite high even under a strong noise level. The reasons for this phenomenon are due to the local temporal features associated with each desired spike. The decision is made by a combination of all the local temporal features. For small number of desired spikes, the neurons make decision based on a relatively less number of temporal features. This small number of features only cover a part range of the whole time window, which inevitably leads to a lower performance compared to a more number of spikes.

In addition, we also use the tempotron rule [13], [24] to perform the same classification task. The tempotron rule is known as an efficient rule for spatiotemporal classification task. Fig. 6 shows the classification performance.

As can be seen from Fig. 6, the PSD rule outperforms the tempotron rule. This is because that the PSD rule makes a decision based on a combination of several local temporal features over the entire time window, but the tempotron rule only makes a decision by firing one spike or not based on one local temporal feature.

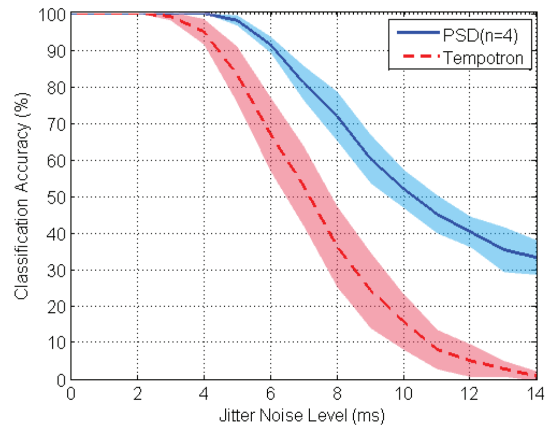

Fig. 6. The classification performance of the PSD rule compared to the tempotron rule.

IV. Conclusion

The PSD rule can perform both the association and the classification task. Through simulations, it can be seen that the neuron can be successfully trained to reproduce spikes as the desired spike train. In the classification task, the neuron is capable to classify different categories with the PSD rule. We have proposed two decision making-schemes which are the absolute confidence and the relative confidence criteria. The classification task is largely improved by the relative confidence criterion. For the classification of spatiotemporal spike patterns, the performance of the PSD rule is better than the tempotron rule. In addition, a sufficient number of desired spikes can also benefit the classification performance of the PSD rule.

References

[1] F. Rieke, D. Warland, Rob, and W. Bialek, *Spikes: Exploring the Neural Code*, 1st ed. Cambridge, MA: MIT Press, 1997.

[2] O. Barak and M. Tsodyks, "Recognition by variance: Learning rules for spatiotemporal patterns," *Neural Computation*, vol. 18, pp. 2343–2358, October 2006.

[3] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, "Sensory neural codes using multiplexed temporal scales," *Trends in Neurosciences*, vol. 33, no. 3, pp. 111–120, Mar. 2010.

[4] M. J. Berry and M. Meister, "Refractoriness and neural precision." *Journal of Neuroscience*, vol. 18, no. 6, pp. 2200–2211, Mar. 1998.

[5] V. J. Uzzell and E. J. Chichilnisky, "Precision of spike trains in primate retinal ganglion cells." *J Neurophysiol*, vol. 92, no. 2, pp. 780–789, Aug. 2004.

[6] P. Reinagel and R. C. Reid, "Temporal coding of visual information in the thalamus," *Journal of Neuroscience*, vol. 20, no. 14, pp. 5392–5400, Jul. 2000.

[7] W. Bair and C. Koch, "Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey." *Neural Computation*, vol. 8, no. 6, pp. 1185–1202, Aug. 1996.

[8] Z. F. Mainen and T. J. Sejnowski, "Reliability of spike timing in neocortical neurons," *Science*, vol. 268, no. 5216, pp. 1503–1506, Jun. 1995.

[9] R. Kempter, W. Gerstner, and J. L. van Hemmen, “Spike-based compared to rate-based hebbian learning.” in *NIPS’98*, 1998, pp. 125–131.
[10] A. Borst and F. E. Theunissen, “Information theory and neural coding.” *Nature Neuroscience*, vol. 2, no. 11, pp. 947–957, Nov. 1999.
[11] J. J. Hopfield, “Pattern recognition computation using action potential timing for stimulus representation,” *Nature*, vol. 376, no. 6535, pp. 33–36, Jul. 1995.
[12] S. M. Bohte, J. N. Kok, and H. La Poutre, “Error-backpropagation in temporally encoded networks of spiking neurons,” *Neurocomputing*, vol. 48, no. 1, pp. 17–37, 2002.
[13] R. Gütig and H. Sompolinsky, “The tempotron: a neuron that learns spike timing-based decisions,” *Nature Neuroscience*, vol. 9, no. 3, pp. 420–428, Feb. 2006.
[14] F. Ponulak, “Resume–new supervised learning method for spiking neural networks,” Institute of Control and Information Engineering, Poznoń University of Technology, Tech. Rep., 2005.
[15] F. Ponulak and A. J. Kasinski, “Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification, and Spike Shifting.” *Neural Computation*, vol. 22, no. 2, pp. 467–510, 2010.
[16] R. V. Florian, “The Chronotron: A Neuron that Learns to Fire Temporally Precise Spike Patterns,” *PloS One*, vol. 7, no. 8, p. e40233, 2012.
[17] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, “SPAN: Spike Pattern Association Neuron for Learning Spatio-Temporal Spike Patterns,” *International Journal of Neural Systems*, vol. 22, no. 04, 2012.
[18] 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.
[19] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
[20] E. M. Izhikevich, “Resonate-and-fire neurons,” *Neural Networks*, vol. 14, no. 6-7, pp. 883–894, 2001.
[21] A. Hodgkin and A. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *Journal of Physiology*, vol. 117, pp. 500–544, 1952.
[22] E. M. Izhikevich, “Simple model of spiking neurons.” *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, November 2003.
[23] M. Rossum, “A novel spike distance,” *Neural Computation*, vol. 13, no. 4, pp. 751–763, 2001.
[24] Q. Yu, H. Tang, K. Tan, and H. Li, “Rapid Feedforward Computation by Temporal Encoding and Learning With Spiking Neurons,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1539–1552, 2013.