An STDP-Based Supervised Learning Algorithm for Spiking Neural Networks

Zhanhao Hu[1], Tao Wang[2], and Xiaolin Hu[3(✉)]

[1] Department of Physics, Tsinghua University, Beijing, China
[2] Huawei Technology, Beijing, China
[3] Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Computer Science and Technology, Center for Brain-Inspired Computing Research (CBICR), Tsinghua University, Beijing, China
xlhu@tsinghua.edu.cn

Abstract. Compared with rate-based artificial neural networks, Spiking Neural Networks (SNN) provide a more biological plausible model for the brain. But how they perform supervised learning remains elusive. Inspired by recent works of Bengio et al., we propose a supervised learning algorithm based on Spike-Timing Dependent Plasticity (STDP) for a hierarchical SNN consisting of Leaky Integrate-and-fire (LIF) neurons. A time window is designed for the presynaptic neuron and only the spikes in this window take part in the STDP updating process. The model is trained on the MNIST dataset. The classification accuracy approach that of a Multilayer Perceptron (MLP) with similar architecture trained by the standard back-propagation algorithm.

Keywords: STDP · SNN · Supervised learning

1 Introduction

Rate-based deep neural networks (RDNN) with back-propagation (BP) algorithm have got great developments in recent years [10]. Neurons in these networks deliver information by floating numbers. But in the brain, signals are carried on by spikes, a kind of binary signals. This property can be captured by a spiking neural networks (SNN). But how the SNNs are trained remains largely unknown.

Several works studying supervised algorithm on SNN have made some progress recently. Some works [2,3,7,8,12,17] make use of time coding by spikes. In a very first work [2], each neuron is only allowed to fire a single spike. The model is then expanded to allowing multiple spikes by later studies [3,7]. Networks in these papers usually need to keep multiple channels with independent weights between two neurons. These channels account for different time delays [2] or order numbers of spikes in the spike train [17]. These algorithms are designed to learn spike trains, but classification on large datasets is hard for these models.

D. Liu et al. (Eds.): ICONIP 2017, Part II, LNCS 10635, pp. 92–100, 2017.
https://doi.org/10.1007/978-3-319-70096-0_10

In fact, the algorithms need to convey real numbers to spike trains. Due to the difficulty of this conversion and recognizing ability of the network, these models can only work on very simple datasets.

Recently, Bengio et al. proposes an idea to build a two-phased learning algorithm for energy-based models called e-prop [14]. They implement the algorithm on an energy-based model with input neurons clamped to input data and output neuron variable under target signals. Neurons are free from target signals in the first phase, and the state of which is denoted by s^0. Dynamics of output neurons are changed slightly by target signals in the second phase, and the state of neurons is s^ξ. Let $\rho()$ represent the active function. The weight W_{ij} of synapse between neuron j and neuron i is updated by

$$W_{ij} \leftarrow W_{ij} + \eta \Delta W_{ij}, \tag{1}$$

where

$$\Delta W_{ij} \propto \lim_{\xi \to 0} \frac{1}{\xi}(\rho(s_i^\xi)\rho(s_j^\xi) - \rho(s_i^0)\rho(s_j^0)). \tag{2}$$

And this rule is a symmetric version of another rule

$$\Delta W \propto \dot{s}_i \rho(s_j), \tag{3}$$

which is studied in previous work [1]. In the work a link has been made between (3) and Spike-Timing Dependent Plasticity (STDP) rule.

STDP rule is thought to be an ideal basis of algorithms on SNN. It is first found in physiological experiment [11], defines that the plasticity of a synapse is only dependent on the time difference of spikes from the two neurons attached by this synapse. But computational significance of STDP is not clear. Several works implement STDP on learning algorithms [6,9,13]. They all take an idea that utilizing the simple property that the synaptic weight is strengthened when the postsynaptic spike is after the presynaptic spike, thus a strict order of presynaptic and postsynaptic spike is needed.

In this work we propose a new STDP-based algorithm on SNN. Also, we find that simply computing all of the spikes using the STDP rule results in poor results. We modify the spike pairs that perform STDP rule, and achieve good results on same benchmark image classification dataset. We stress that we do not change the original STDP rule on single pair of spikes, but provide a way that how to use the STDP rule.

2 Method

2.1 The Network

The network is a bidirectionally connected network with asymmetric weights based on the leaky integrate-and-fire (LIF) neuron model [5]. The state of neuron i is described by membrane potential V_i. The dynamics of V_i is:

$$\tau_V \frac{\mathrm{d}V_i}{\mathrm{d}t} = -V_i + E_L - r_m \sum_{j \in \Gamma_i} \bar{g}_{s,ij} P_{s,j}(V_i - E_{s,j}) + R_m I_e, \tag{4}$$

where I_e is input current, E_L is equilibrium potential, $E_{s,j}$ is determined by types of neurotransmitter, τ_V is a time constant, and R_m is a resistance constant. Γ_i is the set of neurons that have synapses to neuron i. The membrane potential V_i triggers the neuron to release a spike when it reaches a threshold V_{th}, and then is reseted to V_{reset} after the spike. $\bar{g}_{s,ij}P_{s,j}$ represents synaptic conductance from neuron j to neuron i, where $\bar{g}_{s,ij}$ represents the maximum strength of the synapse, and $P_{s,j}$ represents the probability of opened neurotransmitter gates. The dynamics of $P_{s,j}$ is

$$\tau_P \frac{\mathrm{d}P_{s,j}}{\mathrm{d}t} = -P_{s,j} + \sum_k \delta(t - T_j^{(k)}). \tag{5}$$

The variable $P_{s,j}$ increases by a unit amount every time neuron j spikes, and decreases to zero spontaneously. $\delta()$ is a Dirac function, which means $\delta(x) = 0$, $(x \neq 0)$ and $\int_{-\infty}^{\infty} \delta(x)\mathrm{x} = 1$. $[T_j^{(1)}, T_j^{(2)}, ...]$ represents for spike train of neuron j.

We set all $E_{s,j}$ to 0 V, and the input current I_e to 0 μA. Also, for the sake of convenience, we write $\bar{g}_{s,ij}$ to W_{ij}, and introduce an input summation for postsynaptic neuron i

$$P_i = \sum_{j \in \Gamma_i} \bar{g}_{s,ij} P_{s,j}, \tag{6}$$

and rewrite the basic dynamics (4) and (5) as

$$\tau_V \frac{\mathrm{d}V_i}{\mathrm{d}t} = -V_i + E_L - r_m P_i (V_i - E_s), \tag{7}$$

$$\tau_P \frac{\mathrm{d}P_i}{\mathrm{d}t} = -P_i + \sum_{j,k} W_{ij} \delta(t - T_j^{(k)}). \tag{8}$$

The network consists of an input layer, a hidden layer, and an output layer. We denote the data for supervised learning by normalized input signal v_x and target signal v_y.

For neuron i in the input layer, we simply let it be controlled by input signal $v_{x,i}$:

$$P_i = P_0 v_{x,i}, \tag{9}$$

and P_0 is a constant to convert the scales. Neurons in the input layer fire in a fixed pattern under an input proportional to input signal. The neurons in the hidden layer are not affected by any signals from data directly, and act according to (7) and (8).

Situation for neurons in the output layer is a bit more complicated. Like the e-prop method [14], learning is performed in two phases, named inference phase and learning phase in this paper. The only difference between the two phases is the dynamics of the output layer neurons. In the inference phase, the neurons also act according to (7) and (8). The network gives an inference result

by counting the frequency of spikes of output neurons in this phase. And in the learning phase, we add an item represents for effect of target signals:

$$\tau_P \frac{\mathrm{d}P_i}{\mathrm{d}t} = -P_i + \sum_{j,k} W_{ij}\delta(t - T_j^{(k)}) + \beta(P_0 v_{y,i} - P_i), \tag{10}$$

where $v_{y,i}$ is the ith target signal and β controls the effectiveness of target signals.

2.2 The Learning Rule

We adopt the original STDP functions. The STDP function represent the relationship of modification δW_{ij} of the synapse W_{ij} from presynaptic neuron j to postsynaptic neuron i, and the firing time of two spikes fire at t_j and t_i respectively. The commonly used exponential form [15] of STDP function can be

$$\delta W_{ij}(t_i, t_j) = f(t_i - t_j) = \begin{cases} e^{-\frac{t_i - t_j}{\tau_m}}, & \text{when } t_i > t_j \\ 0, & \text{when } t_i = t_j \\ -e^{-\frac{t_j - t_i}{\tau_m}}, & \text{when } t_i < t_j \end{cases}. \tag{11}$$

And we can also use a sinusoidal form [16] as

$$\delta W_{ij}(t_i, t_j) = f(t_i - t_j) = \begin{cases} \sin(\frac{t_i - t_j}{\tau_w}\pi), & \text{when } \Delta t \in [-\tau_w, \tau_w] \\ 0, & \text{otherwise} \end{cases}, \tag{12}$$

where τ_w is a time constant. The two functions are plotted in Fig. 1. During the experiment we found that the sinusoidal form resulted in better consequence, so all results presented in the paper are based on (12).

The STDP rule is implemented on a time window in the learning phase after the inference phase. We find that simply summing up all of the spike pairs in a bidirectional network does not work. When the STDP function is approximately anti-symmetric which means $f(\delta t) = -f(-\delta t)$, we have

$$\Delta W_{ij} = \sum_{t_i, t_j} f(t_i - t_j) = -\sum_{t_i, t_j} f(t_j - t_i) = -\Delta W_{ji}. \tag{13}$$

It means that the synapse modifications in two directions of two neurons are always opposite. Consider a situation that two neurons' firing rates are increasing in a same mode, so that average modification of the two synapse are expected to be symmetric, which is $\Delta W_{ij} = \Delta W_{ji}$. Along with (13), we have $\Delta W_{ij} = \Delta W_{ji} = 0$. This makes no sense for learning and implementing this operation can not learn the model well.

For breaking this symmetry we made a slight modification. We redefined the rules of multiple spikes in a time window $[0, T]$. That is, for synapse W_{ij}, spikes

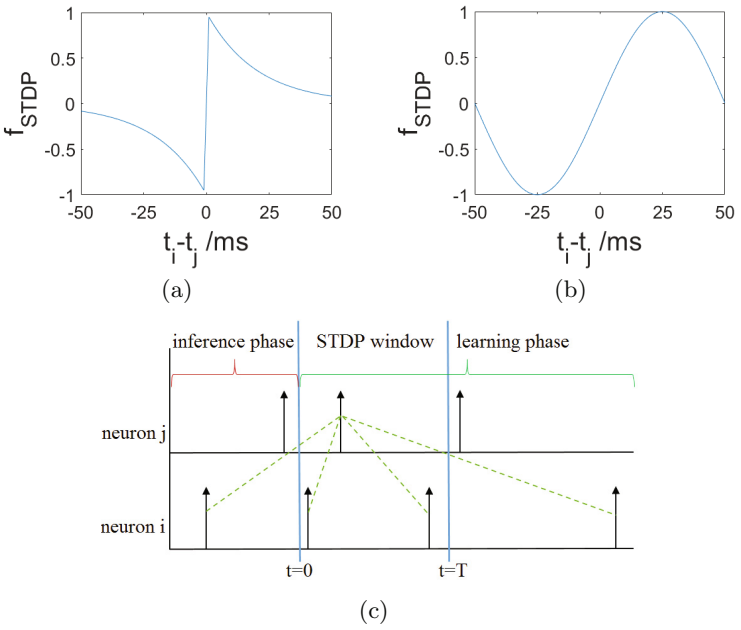


Fig. 1. (a) The exponential form of STDP function. (b) The sinusoidal form of STDP function. (c) Illustration of details of implementing STDP. The learning window is embedded in the learning phase, which is the space between the two vertical lines. Only spike pairs indicated by dotted lines are taken into consider for updating W_{ij}, which is the synaptic weight from presynaptic neuron j to postsynaptic neuron i.

fired by presynaptic neuron j only in time window $[0, T]$, and spikes fired by postsynaptic neuron i in time window $[-\infty, \infty]$ are taken into account:

$$\Delta W_{ij} \propto \int_0^T \mathrm{d}t_j \int_{-\infty}^{\infty} \mathrm{d}t_i f(t_j - t_i) \sum_{k,l} \delta(t_i - T_i^{(k)}) \delta(t_j - T_j^{(j)}). \tag{14}$$

Because of the local property of STDP rule, which means only spikes that the time distance is not larger than τ_w in (12) actually effect, the scope of spikes fired by postsynaptic neuron is $[-\tau_w, T + \tau_w]$ in fact.

In fact, when STDP rule is implemented on time window $[0, T]$, it means the STDP is somehow "turned on" at the time $t = 0$ and "turned off" at the time $t = T$. And more specifically, STDP can be considered as a consequence of some kinds of biochemical signals from both presynaptic neuron and postsynaptic neuron [4]. We propose an idea that STDP is considered to be "turned on" by activating the production or transmission of the biochemical signal triggered by presynaptic neuron spikes, and also it is "turned" off by suppressing these signals, while signals related to postsynaptic spikes are existed all time along.

The learning algorithm is summarized in Algorithm 1.

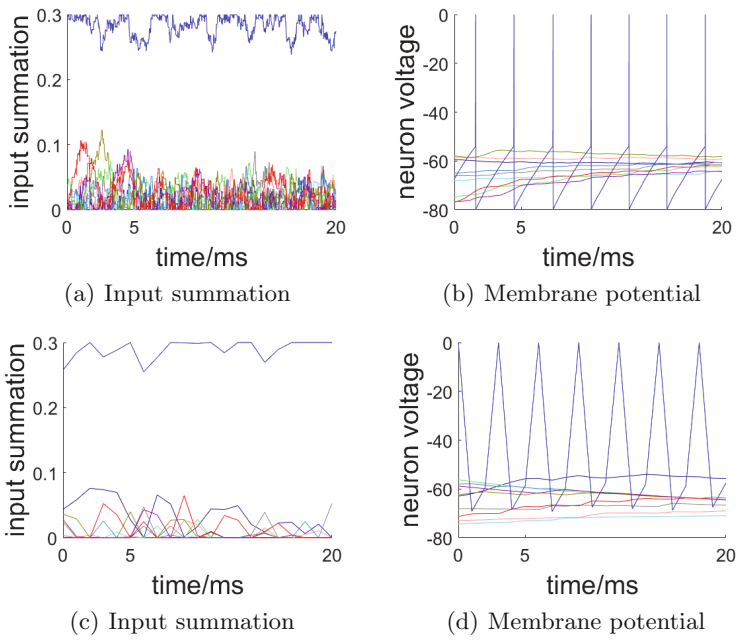
Algorithm 1. Training the spiking neural network

1: Simulate an inference phase in the time window of $[-t_0, 0]$ and record the spike train $[T_i^{(1)}, T_i^{(2)}, ..., T_i^{(m_i)}]$.
2: Simulate a learning phase in the time window of $[0, T + t_w]$ and record the spike train $[T_i^{(m_i+1)}, T_i^{(m_i+2)}, ..., T_i^{(M_i)}]$.
3: $W_{ij}^{(n+1)} = W_{ij}^{(n)} + \alpha \int_0^T \mathrm{d}t_j \int_{-\infty}^{\infty} \mathrm{d}t_i f(t_j - t_i) \sum_{k,l} \delta(t_i - T_i^{(k)}) \delta(t_j - T_j^{(j)})$.

3 Results

We implement the model on the MNIST dataset. The dataset contains 60,000 training images and 10,000 test images. And the images are in gray scale and have size 28×28. The size of the network is 784-200-10, which indicates the numbers of neurons in input layer, hidden layer, and output layer, respectively.

We use the Euler method to approximate the differential function (7) and (8). Figure 2 is the simulation illustration of input summation and the membrane potential of 10 output layer neurons with different time step. We set $\tau_V = 20\,\mathrm{ms}$,



(a) Input summation (b) Membrane potential

(c) Input summation (d) Membrane potential

Fig. 2. Simulation illustration of input summation and membrane potential of 10 output layer neurons in an inference phase. The synaptic weights have been trained on MNIST dataset. Only the 8th neuron have a maximal input, and fires in a maximal pattern, while other neurons are not. The simulation for (7) and (8) is processed using Euler method. We have tried different time steps, as 0.01 ms for (a)(b) and 1 ms for (c)(d)

$\tau_P = 10\,\text{ms}$, $E_s = 0$, $E_L = -70\,\text{mV}$, $V_{reset} = -80\,\text{mV}$, $V_{th} = -54\,\text{mV}$, and a hard bound $[0, 0.3]$ for P_i. In fact, we find that a simulation time step of 1 ms is enough to depict the spiking trains, so we use a step of 1 ms in our later experiment.

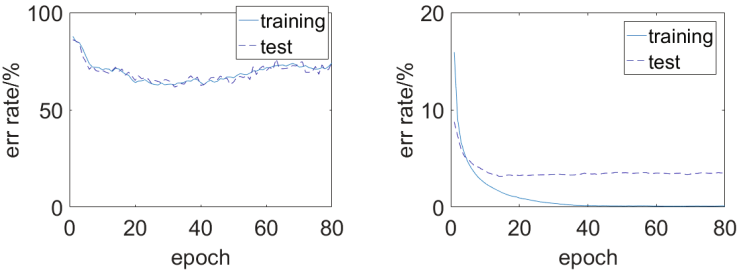


Fig. 3. The error rates against epochs on the MNIST dataset. (a) Simply take all of the spike pairs into account. (b) Use the proposed method.

We test our model on the MNIST dataset (Fig. 3). Using the STDP rule with all of the spikes in the time window taken in to account did not work. By using the proposed method, the error rate on training set is able to decrease to 0.0% in the experiment, which proves the convergence of algorithm experimentally. For comparison, we also implement the e-prop and MLP which have similar architecture to our model (the same number of input, hidden and output neurons). Several other STDP-based algorithms are also compared. The test accuracies on the MNIST dataset are summarized in Table 1. The test accuracy of our method is greater than other STDP-based algorithms, except for the algorithm that use a convolutional architecture [9].

Table 1. Comparison of different algorithms

Model	Neural coding	Test accuracy/%
E-prop	Rate-based	97.5
MLP	Rate-based	98.5
Two layer network [13]	Spike-based	93.5
Two layer network [6]	Spike-based	95.0
Convolutional SDNN [9]	Spike-based	98.4
Proposed model	Spike-based	96.8

4 Discussion

We describe an STDP-based supervised learning algorithm on SNN, and get good results on the MNIST classification task. The accuracy approaches that of an MLP with a similar architecture, which indicates the effectiveness of this algorithm. Compared with existing algorithms for training SNNs, the proposed algorithm have achieved competing results.

The algorithm suggests that biological neurons may not modify their synapses under the STDP rule all the time. STDP takes effect only when the supervisory signals are applied. In addition, the algorithm suggests that not all spikes of the presynaptic neuron participate in the STDP learning process for the synapse. Instead, there may exist a time window and only the spikes during this window should be counted. But biochemical evidence is needed to validate these predictions.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant 91420201, Grant 61332007, Grant 61621136008 and Grant 61620106010, in part by the Beijing Municipal Science and Technology Commission under Grant Z161100000216126, and in part by Huawei Technology under Contract YB2015120018.

References

1. Bengio, Y., Mesnard, T., Fischer, A., Zhang, S., Wu, Y.: STDP as presynaptic activity times rate of change of postsynaptic activity. arXiv preprint (2015). arXiv:1509.05936
2. Bohte, S.M., Kok, J.N., La Poutre, H.: Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing **48**(1), 17–37 (2002)
3. Booij, O., tat Nguyen, H.: A gradient descent rule for spiking neurons emitting multiple spikes. Inf. Process. Lett. **95**(6), 552–558 (2005)
4. Clopath, C., Büsing, L., Vasilaki, E., Gerstner, W.: Connectivity reflects coding: a model of voltage-based STDP with Homeostasis. Nat. Neurosci. **13**(3), 344–352 (2010)
5. Dayan, P., Abbott, L.F.: Theoretical Neuroscience, vol. 806. MIT Press, Cambridge (2001)
6. Diehl, P.U., Cook, M.: Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Front. Comput. Neurosci. **9**, 99 (2015)
7. Ghosh-Dastidar, S., Adeli, H.: A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. Neural Netw. **22**(10), 1419–1431 (2009)
8. Gütig, R., Sompolinsky, H.: The tempotron: a neuron that learns spike timing-based decisions. Nat. Neurosci. **9**(3), 420–428 (2006)
9. Kheradpisheh, S.R., Ganjtabesh, M., Thorpe, S.J., Masquelier, T.: STDP-based spiking deep neural networks for object recognition. arXiv preprint (2016). arXiv:1611.01421
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)

11. Markram, H., Lübke, J., Frotscher, M., Sakmann, B.: Regulation of synaptic efficacy by coincidence of postsynaptic APS and EPSPS. Science **275**(5297), 213–215 (1997)
12. Ponulak, F., Kasiński, A.: Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. Neural Comput. **22**(2), 467–510 (2010)
13. Querlioz, D., Bichler, O., Dollfus, P., Gamrat, C.: Immunity to device variations in a spiking neural network with memristive nanodevices. IEEE Trans. Nanotechnol. **12**(3), 288–295 (2013)
14. Scellier, B., Bengio, Y.: Equilibrium propagation: bridging the gap between energy-based models and backpropagation. Front. Comput. Neurosci. **11** (2017). Article no. 24
15. Song, S., Miller, K.D., Abbott, L.F.: Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nat. Neurosci. **3**(9), 919–926 (2000)
16. Xie, X., Seung, H.S.: Spike-based learning rules and stabilization of persistent neural activity. In: Advances in Neural Information Processing Systems, pp. 199–208 (2000)
17. Xie, X., Qu, H., Yi, Z., Kurths, J.: Efficient training of supervised spiking neural network via accurate synaptic-efficiency adjustment method. IEEE Trans. Neural Netw. Learn. Syst. **28**(6), 1411–1424 (2017)