Neurocomputing 275 (2018) 94–106

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Relative ordering learning in spiking neural network for pattern recognition

Zhitao Lin[a], De Ma[b,*], Jianyi Meng[c], Linna Chen[a]

[a] Institute of VLSI Design, Zhejiang University, Hangzhou 310027, Zhejiang, China
[b] College of Computer Science, Zhejiang University, Hangzhou, 310027, China
[c] State Key Laboratory of ASIC and System, Fudan University, Shanghai 201203, China

ARTICLE INFO

Article history:
Received 19 July 2016
Revised 23 February 2017
Accepted 8 May 2017
Available online 13 May 2017

Communicated by M. Bianchini

Keywords:
Pattern recognition
Relative ordering learning
Spiking neural network
Supervised learning
Potentiation and depression hybridized adaptation

ABSTRACT

The timing of spikes plays an important role in the information processing of brain. However, for temporal-based learning algorithms, the temporally precise spike as learning target is not able to fit with the variety of stimuli. The performance of spiking neural networks (SNNs) is adversely affected by fixed target spikes. This paper proposes a new learning rule called relative ordering learning (ROL) for SNNs. In contrast to the existing algorithms that drive a neuron to fire or not fire or to fire at precise times, the ROL rule trains the neuron to emit the first spike to respond to the associated stimuli. Without the limitation of specified precise firing times, the ROL rule can adjust output spikes to fit with the feature of input stimuli dynamically and adaptively. Our experiments have demonstrated the great generalization ability of the ROL rule and its robustness against noise. With a few neurons and ultra-low training cost, the network trained by the ROL rule achieved the classification accuracies of 98.1% and 93.8% for training on the Iris and MNIST datasets, respectively, and the accuracies for testing are 94.2% and 90.3%. The comparison with MLP network also shows the great attraction of the SNN trained by the ROL rule as a solution for fast and efficient recognition tasks.

1. Introduction

As the third generation of the artificial neural network (ANN), the spiking neural network (SNN) is more biologically plausible and more computationally efficient [1]. Spikes act as the medium of information processing in the SNN; however, the means by which information is represented with spikes in the brain is still unclear. Experiments have shown that neurons are able to precisely react to stimuli on a millisecond timescale, especially in the retina [2,3] and the visual cortex [4]. The results of these experiments also support the hypothesis of temporal coding [5], which suggests that the precise timing of a spike conveys information. This theory has become one of the most widely studied neural coding schemes. Based on temporal coding, researchers have developed many effective learning algorithms to train spiking neurons. The SpikeProp rule is one of the first supervised learning algorithms for the SNN [6]. By training the neuron to fire at a desired, precise time, it is capable of solving nonlinear classification tasks. Some modified forms of SpikeProp suggest that conditioning one neuron can lead to the generation of a desirable multi-spike train [7]. Derived from the Widrow–Hoff rule, the ReSuMe rule applies to the process of training neurons based on a learning window similar to spike-timing-dependent-plasticity (STDP) [8] in order to minimize the error between the target spike train and the actual output [9]. Other learning algorithms such as SPAN [10], PSD [11] and Chronotron [12] have also been proposed to train neurons to precisely emit a single spike or a spike train. In all of these algorithms, the interpretation of the network's output spikes depends on their exact timing. With abundant information available from temporal coding, the SNN has the potential to learn to deal with complex, real-world stimuli.

For the temporal-based algorithms, some specific spiking times need to be designated as learning targets of neurons to distinguish between categories of stimuli. Currently, these algorithms do not provide any methods or guidelines for designing appropriate firing times. Learning targets greatly affects the performance of learning algorithms [7,13]. Afferent spatiotemporal spike patterns belong to different categories and have their own inherent data structure, and the learning target specified by researchers is likely incompatible with that. The improperly corresponding relationship between the structure of spike pattern and the learning target can reduce

* Corresponding author.
E-mail addresses: 11310050@zju.edu.cn (Z. Lin), made@hdu.edu.cn (D. Ma), mengjy@fudan.edu.cn (J. Meng), 21510096@zju.edu.cn (L. Chen).

http://dx.doi.org/10.1016/j.neucom.2017.05.009
0925-2312/

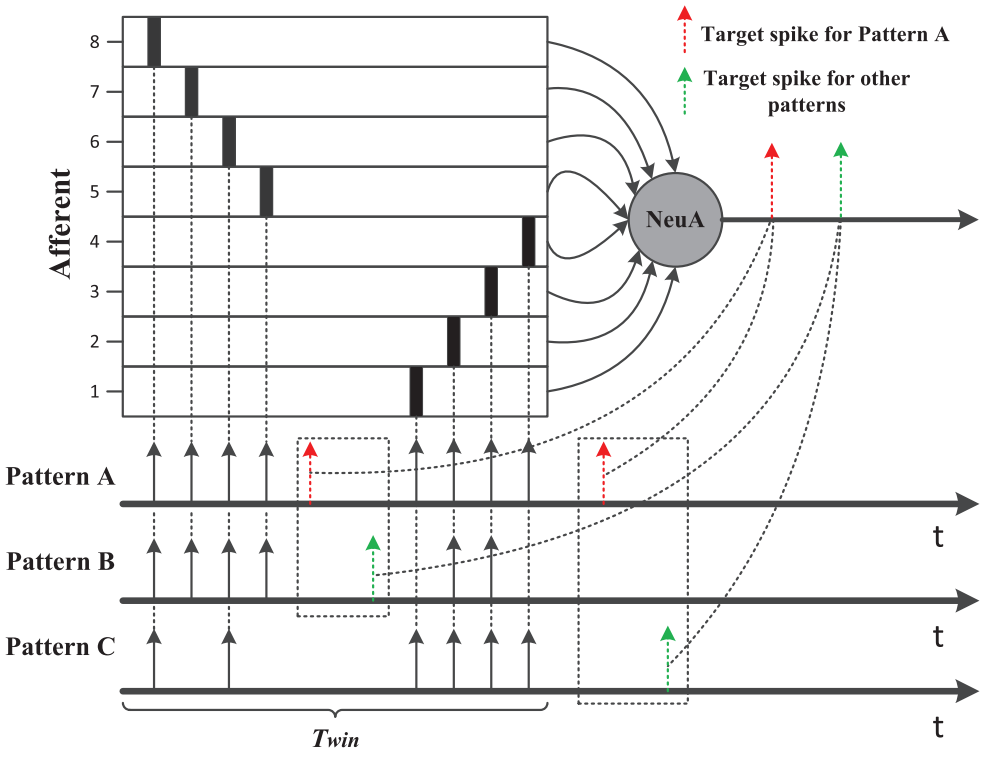

Fig. 1. The deficiency of learning rules based on precise single spike. The solid arrow lines on time axis represent input spike patterns. It is improper that the target spikes for Pattern A and Pattern B are assigned within the encoding time window, while setting target spikes outside the encoding time window might lose the information of early spikes.

the robustness of the SNN and even make the training process not convergent.

As an illustration, Fig. 1 shows an example that reveals the deficiency of rules based on temporal coding. The identity of patterns is encoded by the timing of a single spike. The neuron associated with Pattern A is trained to fire at a certain time for Pattern A and to fire at a different time for other patterns. The first half of the afferent spikes in Pattern A and Pattern B are identical. If target spikes are assigned within the encoding time window as Fig. 1 shows, the latter part of the spikes is ignored as training continues, and these two patterns become the same for the learning neuron. The neuron is consequently not able to distinguish Pattern A from Pattern B. Avoiding this situation with real-world, complex stimuli is difficult unless target spikes are assigned outside the encoding time window or a pre-analysis is performed to design proper firing times. A pre-analysis makes the recognition system more complex and inefficient. When setting target spikes outside the encoding time window, learning neurons receive complete information about patterns passively. Redundant information frequently increases the training difficulty. Moreover, leakage neuron models are widely used in SNNs [14,15]. With target spikes outside the encoding time window, early spikes have little effect on generating target spikes when the time window is too long, especially for a single-layer SNN. In this situation, distinguishing Pattern A from Pattern C, as shown in Fig. 1, becomes difficult because of information loss from early spikes.

As for spike train learning, it is more difficult to design proper target spike trains for real-world stimuli. Emitting a multi-spike train has requirements relative to the number and the distribution of afferent spikes. To obtain appropriate performance of the SNN trained by spike train learning rules, stimuli need to be encoded by a phase code or other complex encoding methods to create distributed spatiotemporal spike patterns [11,15]. Additionally, the decoding of output spikes typically requires measurement of the

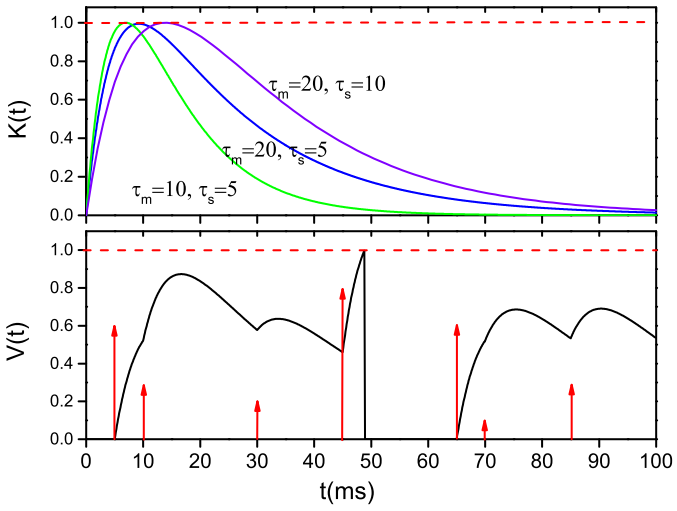

Fig. 2. The kernel function and the membrane potential trace of neuron ($V_{thr} = 1.0$). The red arrows represent afferent spikes and their length reflects the weight of the corresponding synapse. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

similarity between the actual efferent spike train and the target spike trains of different categories [11,12,15]. This increases the complexity of the SNN recognition system and reduces the system's efficiency.

In contrast to precise spiking times, the relative ordering of spikes is more concise and more robust. Rank order coding is used to encode information with relative timing of spikes [16,17]. This process only focuses on the order of input spikes, in which pre-order spikes take on a greater role in the membrane potential of a neuron. With the advantage of fast learning, rank-order based learning rules are well studied and utilized to train the evolving spiking neuron network (eSNN) [18,19]. Recent research on mammalian retinal ganglion cells also confirms the

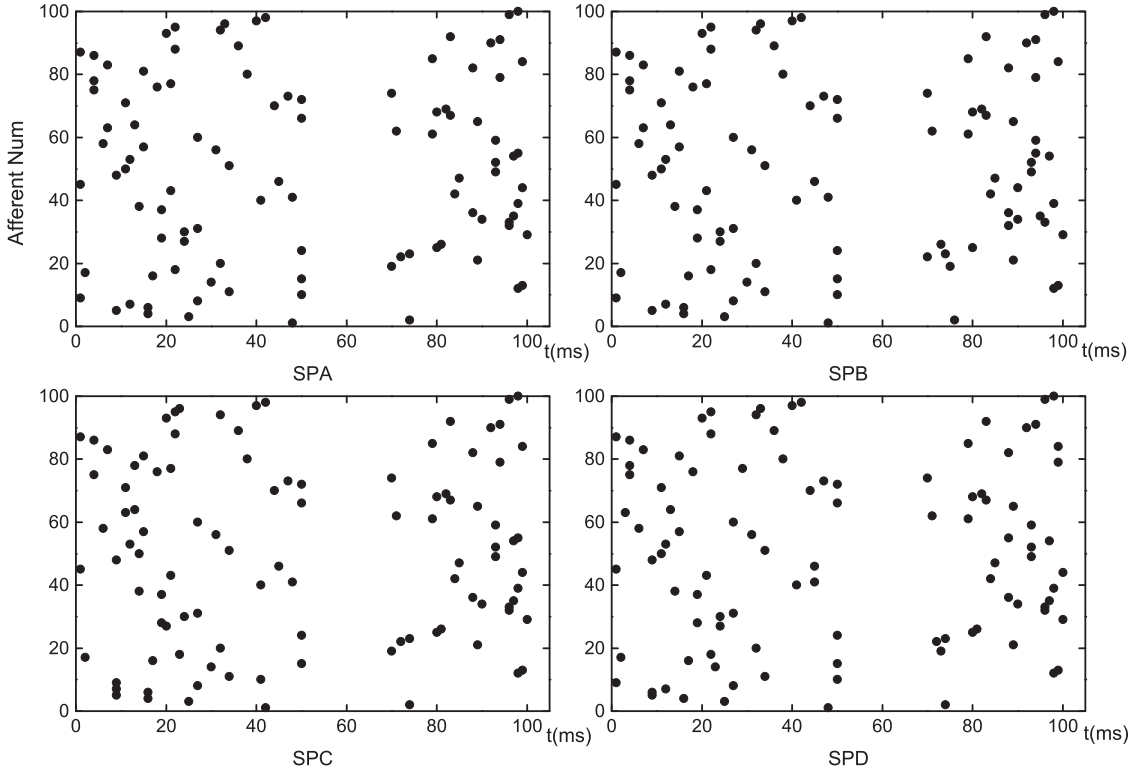

Fig. 3. Illustration of spike patterns.

Table 1
Comparison of training algorithms on the Iris dataset.

Algorithm	Inputs	Hidden	Outputs	Iterations	Training set	Test set
ASA [15]	48	4	3	2	96%	95%
GPSNN [33]	20	6	3	97	97.4% ± 1.1	96.5% ± 1.5
GMSES [7]	21	8	1	241	99.9%	94.7%
SpikeProp [34]	17	8	3	37	≥ 95%	92.7%
SWAT [35]	16	208	3	500	95.5% ± 0.6	95.3% ± 3.6
Tempotron [28]	48	–	3	< 100	99.6% ± 0.8	92.5% ± 3.3
ROL (this paper)	48	–	3	4	98.1% ± 0.1	94.2% ± 1.1

Table 2
Correct rate of each digit for the test set.

	G0	G1	G2	G3	G4	G5	G6	G7	G8	G9	CR_D (%)
D0	956	2	2	1	1	5	6	2	3	1	97.6
D1	1	1098	2	2	1	3	4	1	22	0	96.7
D2	10	7	863	10	12	4	20	11	71	8	83.6
D3	8	0	14	878	4	33	9	11	32	9	86.9
D4	2	2	1	0	925	1	11	0	10	27	94.2
D5	10	3	1	25	11	757	23	6	43	6	84.9
D6	11	3	2	1	8	13	902	0	4	0	94.2
D7	2	11	19	8	13	2	0	931	4	32	90.6
D8	17	7	2	12	15	23	15	5	861	8	88.4
D9	11	7	2	15	59	10	0	21	18	861	85.3
CR_G (%)	93.0	96.3	95.0	92.2	88.2	88.9	91.1	94.2	80.6	90.4	

Table 3
Comparison of SNNs on the MNIST dataset.

Network-type	Neurons (Exclude input layer)	Synapses	Preprocessing	Training samples	Test samples	Performance (%)
Dendritic neurons [39]	10 (2000 dendrites)	50,100	Thresholding	10,000	5000	90.26
Spiking RBM [40]	1010	647,000	None	120,000	10,000	94.09
Four layer network [41]	70,242	≈ 133 million	Orientation detection	2000	1000	91.64
Spiking NN [38]	2410	≈ 2 million	None	60,000	10,000	98.6
Three layer network (inhibitory layer)[36]	12,800	≈ 5 million	None	60,000	10,000	95.0
Two layer network (this paper)	100	50,000	None	10,000	10,000	90.34

Table 4
Comparison between the SNN and MLP.

Network	Synapses	Performance (%)	Adds	Multiplications
MLP(784-90-10)	71,560	95.5	71,460(1.0)	71,460(1.0)
SNN ($\Delta t = 0.5$ ms)	50,000	90.33	122,000(1.70)	44,000(0.61)
SNN ($\Delta t = 1.0$ ms)	50,000	90.38	111,000(1.55)	22,000(0.30)
SNN ($\Delta t = 2.0$ ms)	50,000	90.29	105,500(1.47)	11,000(0.15)

$T_{sim} = 110$ ms.

important role of rank order coding [20,21]. In particular, the first stimulus-evoked spikes reliably encode and rapidly transmit information about stimuli [21,22]. Experiments on the IT cortex of monkeys have also demonstrated that the first spikes over a short time window (approximately 12.5 ms) make the greatest contribution in recognition tasks [23]. The experiments suggest that the first spike among neurons carries the most information. Based on these discoveries in biology, the time-to-first-spike model has been used to construct ultra-fast recognition systems for tactile and visual stimuli [24–26]. In [27], the proposed learning rule combines temporal coding and the first-spike-based decision mechanism, which enables rapid classification of input patterns. In fact, the relative ordering of output spikes is often employed to read out the information of afferent patterns in many SNN recognition systems [6,28].

In this paper, we propose a new learning rule called relative ordering learning (ROL). Inspired by the importance of the first spike in the brain, the ROL rule trains a neuron to fire faster than other neurons when its associated spike pattern is presented. Existing learning algorithms propose training a neuron to fire or not fire or to fire at precise times [7,10,11,29]. Alternatively, the ROL rule focuses only on the relative ordering of output spikes. The identity of input patterns is determined by the winning neuron that emits the first spike, which is also called FSMD (First-Spike-Makes-Decision). With the order of output spikes acting as an indicator, the ROL rule adjusts firing times of neurons by different methods based on stochastic gradient descent. Long-term depression (LTD) would be applied to the neurons that wrongly respond to the unassociated patterns, but a compensation mechanism ensures their sufficient sensitivity to associated patterns. The neuron that is expected to fire first would carry out long-term potentiation (LTP) on synapses according to the ROL rule. To the best of our knowledge, this is the first attempt to use the order information of output spikes to guide the training process directly.

The SNN trained by the ROL rule is biologically plausible and has the ability of fast neural information processing. Without the limitation of specified precise spiking times, the ROL rule can adjust output spikes to fit with the feature of input patterns dynamically and adaptively. The combination of flexible output spikes and the FSMD mechanism improves the strength and the adaptability of the SNN in pattern recognition tasks. Moreover, with the ROL rule, we do not need to take a pre-analysis for stimuli to choose proper target spikes. Lastly, the readout of the SNN is also much simpler compared with the similarity measure method used by spike train learning rules.

In the following sections, the details of the ROL rule are presented. Some experiments are performed to demonstrate the learning ability of the ROL rule and its robustness against noise. We also investigate its great generalization ability on real-world stimuli and advantages with respect to the traditional ANNs. Finally, we draw conclusions based on our experimental data and discuss future research.

2. Relative ordering learning rule

In this section, we first describe the spiking neuron model used in the ROL rule. Then, the details of the ROL rule are presented.

2.1. Spiking neuron model

Researchers have developed several kinds of spiking neuron models, such as the Hodgkin–Huxley model [30], the resonate-and-fire model [31], the Izhikevich model [32] and the leaky integrate-and-fire model (LIF) [14]. The LIF model is widely used because it considers both biological plausibility and computational efficiency. In this paper, the LIF model used in [29] is considered. The membrane potential of a neuron is the weighted sum of postsynaptic potentials (PSPs) from all afferent spikes, described as:

$$V(t) = \sum_i w_i \sum_{t_i} K(t - t_i) + V_{rest} \tag{1}$$

w_i is the weight of the ith afferent synapse, and t_i is the arriving time of the spike on this synapse. V_{rest} is the resting potential of the neuron, usually defined as 0. When the membrane potential $V(t)$ exceeds a constant threshold V_{thr}, the neuron emits a spike and its membrane potential is reset to V_{rest}. $K(t)$ is the kernel function shaping PSP:

$$K(t) = \begin{cases} V_{nor}(e^{-t/\tau_m} - e^{-t/\tau_s}) & t > 0 \\ 0 & t \le 0 \end{cases} \tag{2}$$

V_{nor} is the normalization value that limits the maximum value of kernel function to 1. τ_m and τ_s are the decay time constants of PSP. The kernel functions with different decay time constants are shown in Fig. 2. In the ROL rule, we only consider the stimuli composed of single spikes on each synapse.

2.2. The ROL rule

According to the ROL rule, the identity of stimuli is encoded by the ordering of output spikes. When a spike pattern is presented to SNN, the neuron associated with this pattern (labelled by NeuA) should emit the first spike, while other neurons are expected to fire at later times or to remain silent. To achieve this training objective, we designed different adaptation methods based on the relative ordering of output spikes for NeuA and other neurons.

If NeuA emits the first spike, its synaptic weights would not change. If NeuA does not fire, it would be compelled to fire as soon as possible under the ROL rule. We have labelled this case the potentiation adaptation (PA) process. Naturally, we should close the gap between the threshold value and the maximal membrane potential NeuA achieved. Thus, the cost function of the single pattern used here is the same as that presented in [29]:

$$C_{PA} = V_{thr} - V(t_{max}), \tag{3}$$

where t_{max} is the time at which the membrane potential reaches its maximal value, $V(t_{max})$. In this way, the synaptic weights of NeuA are adjusted according to the follow equation:

$$\Delta w_i = -\lambda_1 \frac{\partial C_{PA}}{\partial w_i} = \lambda_1 K(t_{max} - t_i), \quad \text{if} \quad t_i < t_{max} \tag{4}$$

where λ_1 is the learning rate. For the PA process, only synapses where a spike is present before NeuA reaches its maximal potential would be strengthened.

If NeuA fires but is not the first one to respond to its associated patterns, the synaptic weights of NeuA should be improved and thus decrease its spiking latency. We call this case enhanced potentiation adaptation (EPA). The cost function of the single pattern is described as:

$$C_{EPA} = t_{spike} \tag{5}$$

t_{spike} denotes the output spike time. From Eq. (5), we get:

$$\Delta w_i = -\lambda_2 \frac{\partial C_{EPA}}{\partial w_i} = -\lambda_2 \frac{\partial t_{spike}}{\partial w_i} = -\lambda_2 \frac{\partial t_{spike}}{\partial V(t)} \frac{\partial V(t)}{\partial w_i}(t_{spike}) \tag{6}$$

λ_2 is the learning rate of the EPA process. From Eq. (1), $\partial V(t)/\partial w_i$ can be computed as:

$$\frac{\partial V(t)}{\partial w_i}(t_{spike}) = K(t_{spike} - t_i) \tag{7}$$

In [6], $\partial t_{spike}/\partial V(t)$ has been deduced approximately as:

$$\frac{\partial t_{spike}}{\partial V(t)}(t_{spike}) = \frac{-1}{\sum_{t_j < t_{spike}} w_j K'(t_{spike} - t_j)} \tag{8}$$

In Eq. (8), $K'(t)$ is the derived function of $K(t)$. Then, the synaptic weight increment in the EPA process can be deduced:

$$\Delta w_i = \lambda_2 \frac{K(t_{spike} - t_i)}{\sum_{t_j < t_{spike}} w_j K'(t_{spike} - t_j)}, \quad \text{if} \quad t_i < t_{spike} \tag{9}$$

Similar with the PA process, only synapses where a spike is present before NeuA fires would be adjusted in the EPA process.

As for other neurons that are unassociated with the input pattern, they are supposed to fire after NeuA. If they do not emit spikes, their synaptic weights do not need to be adjusted. As long as they fire to respond to the unassociated patterns, LTD is imposed on those synapses that contribute to firing, and thus, their spiking times are delayed. However, the firing behavior of these neurons suggests that there are different degrees of similarity between the patterns associated with them and the input pattern associated with NeuA. Only LTD adaptation possibly prevents these neurons from responding to their associated patterns. Since the initial part of the input pattern incorrectly invokes these neurons' firing, we can assume that the latter part of afferent spikes that arrive after firing times contain the crucial differences in the information about stimuli. LTP should be imposed on the synapses associated with these latter parts of the spikes. By LTD and LTP adaptations on different synapses, these neurons tend to emit spikes at later times for unassociated patterns and they still maintain a certain degree of sensitivity to their own correct patterns. With LTP and LTD cooperating with each other, the ROL rule provides the learning methods necessary to find the essential differences among patterns by adjusting neurons' firing times dynamically. We call this case potentiation and depression hybridized adaptation (PDHA). Specially, for LTD operation, the cost function is shown as:

$$C_{HA}^{d} = -t_{spike} \tag{10}$$

Compared with Eq. (5), we can get:

$$\Delta w_i = -\lambda_2 \frac{K(t_{spike} - t_i)}{\sum_{t_j < t_{spike}} w_j K'(t_{spike} - t_j)}, \quad \text{if} \quad t_i < t_{spike} \tag{11}$$

From Eq. (11), we can see that only the synapses associated with pre-firing spikes are depressed. As for LTP, the anti-STDP process [9] is implemented to adapt synapses that cause spikes after the neuron fires. We choose the kernel function $K(t)$ as the adjusting window here and get the following formula:(λ_3 is the potentiation rate.)

$$\Delta w_i = \lambda_3 K(t_i - t_{spike}), \quad \text{if} \quad t_i > t_{spike} \tag{12}$$

Thus, we adjust the synapses of the neurons that incorrectly fire to respond to the unassociated patterns according to the following formula:

$$\Delta w_i = \begin{cases} -\lambda_2 \frac{K(t_{spike} - t_i)}{\sum_{t_j < t_{spike}} w_j K'(t_{spike} - t_j)} & t_i < t_{spike} \\ \lambda_3 K(t_i - t_{spike}) & t_i > t_{spike} \end{cases} \tag{13}$$

3. Experimental results

In this section, several experiments are presented to demonstrate the performance of the ROL rule. The first experiment intuitively shows the learning process of neurons in a pattern recognition task. The second experiment illustrates the robustness of the ROL rule. Finally, the non-linearly separable Iris dataset and the MNIST dataset are used to demonstrate how the application of the ROL rule can help classify real-world stimuli. A comparative experiment with MLP network is also performed to show the great attraction of the SNN trained by the ROL rule as a solution for fast and efficient recognition tasks.

3.1. The adaptive learning process of neurons

This experiment is devised to demonstrate the pattern recognition ability of the ROL rule preliminarily. In this experiment, learning neurons are connected with n encoding neurons, and each encoding neuron emits a maximum of one spike within the encoding time window T_{win}. We set $n = 100$, $T_{win} = 100$ ms here. Four special spike patterns (SPA, SPB, SPC, SPD) shown in Fig. 3 are constructed. Afferent spikes in SPA are randomly generated with a uniform distribution, and only 10% of spikes were different between SPA and other patterns. Specifically, the different spikes in SPB are concentrated in the latter part of the encoding time window while the latter part of SPA and SPC are identical. In SPD, the different spikes were randomly specified in the whole encoding time window. Additionally, a time vacuole exists between the first part and the latter part of spikes in these patterns. These features of patterns increase the difficulty of classification. Each learning neuron corresponds with a pattern, denoted as NeuA, NeuB, NeuC and NeuD, and their initial synaptic weights are generated from a uniform distribution with short intervals. These four neurons comprise a fully connected single-layer SNN. During the experiment, four patterns are randomly presented to the SNN and synaptic weights adaptation is carried out at the end of the pattern presentation. All 40 patterns are treated as a learning epoch.

For these four patterns, learning rules based on temporal coding were inefficient. By setting a target spike outside the encoding time window, neurons receive the same spikes from SPA and SPC near the target firing time. Thus,it is difficult to distinguish SPA from SPC for precise single spike learning rules. Fig. 4 shows the learning process of neurons trained by the SpikeProp [6]. The target firing times of neurons are set at 102 ms for the associated pattern and at 110 ms for other patterns. We found that SPA could not be recognized by neurons completely. Three neurons fire at nearly the same time for SPA. The situation becomes more complicated when the time vacuole becomes longer. As for spike train learning rules, the design of target spike train should be carefully executed to avoid a case in which multiple target spikes are assigned near the time vacuole. In other words, we need to thoroughly understand prior knowledge about the distribution of afferent spikes to obtain an appropriate target spike train. The essential problem is that the fixed spike times as learning targets do not fit with various patterns in the real world. With the FSMD principle as judgement criterion, the ROL rule can be applied to solve these problems extensively by adjusting neurons' firing time adaptively.

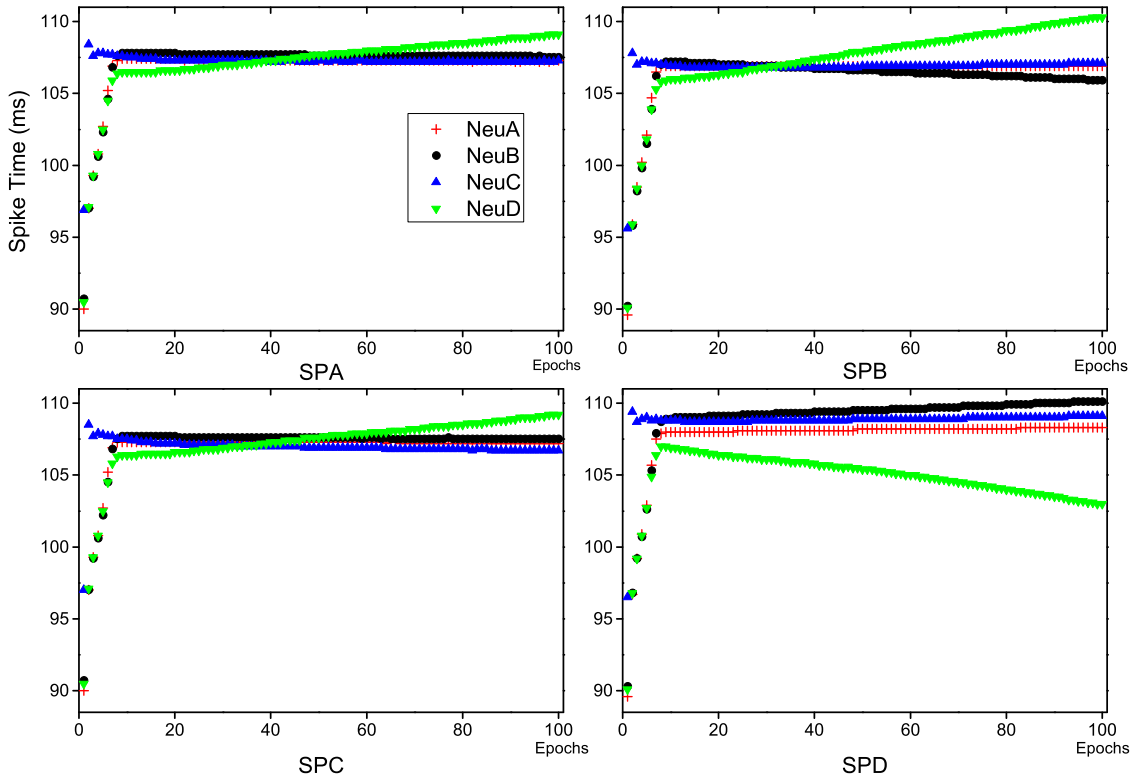

Fig. 4. Distribution of neurons' output spikes under the training of the SpikeProp rule.

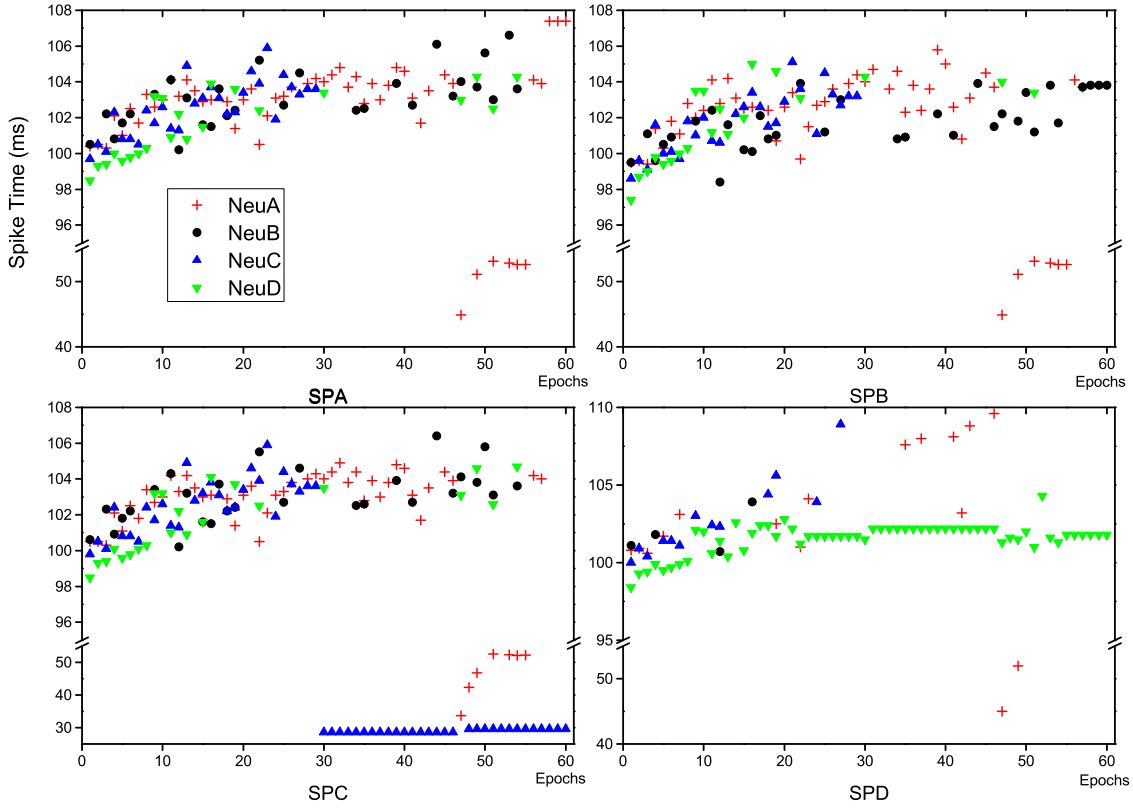

Fig. 5. Distribution of neurons' output spikes under the training of the ROL rule.

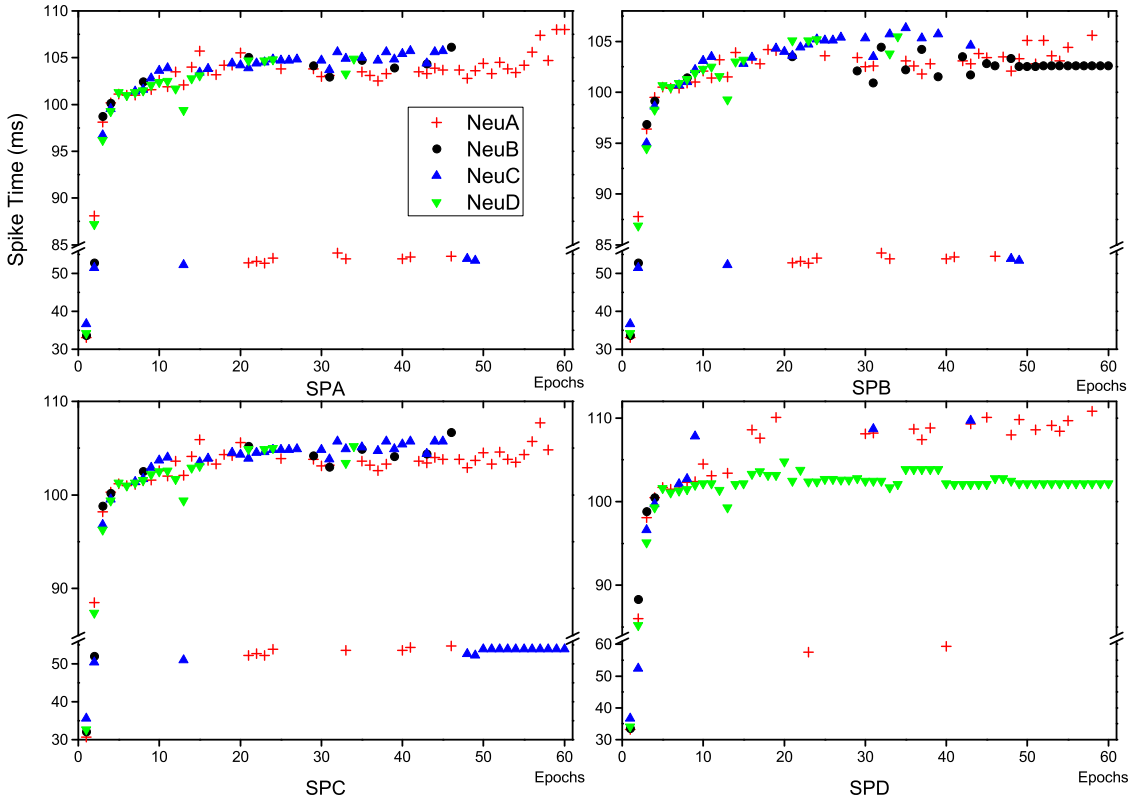

Fig. 6. Distribution of neurons' output spikes under the training of the ROL rule. The learning neurons were assigned with larger initial weights.

Fig. 5 shows the learning process of neurons trained by the ROL rule. In the initial phase, neurons emit spikes inaccurately because the patterns are similar to one another. As training goes on, neurons learn to fire earlier for the corresponding associated pattern while they fire at later times or remain silent for other patterns. The adjustment process of neurons' firing times fully reflects the flexibility and strong adaptability of the ROL rule. NeuA automatically delays its firing time to avoid the confusion caused by the same spikes in SPA and SPB. Because the first part of the spikes is distinctively different between SPC and other patterns, NeuC successfully learns to fire within the encoding time window for SPC. It also proves the rapid classification ability with ROL rule application.

The ROL rule is adaptable to different initial synaptic weights. Fig. 6 shows the results of an experiment with larger initial weights. In contrast to Fig. 5, neurons emit spikes within the encoding time window for multiple patterns at the beginning of training. After approximately 50 epochs of training, all patterns can be recognized correctly under the guidance of FSMD principle. Through further training, the SNN reaches a steady state in which neurons only fire for their respective associated pattern.

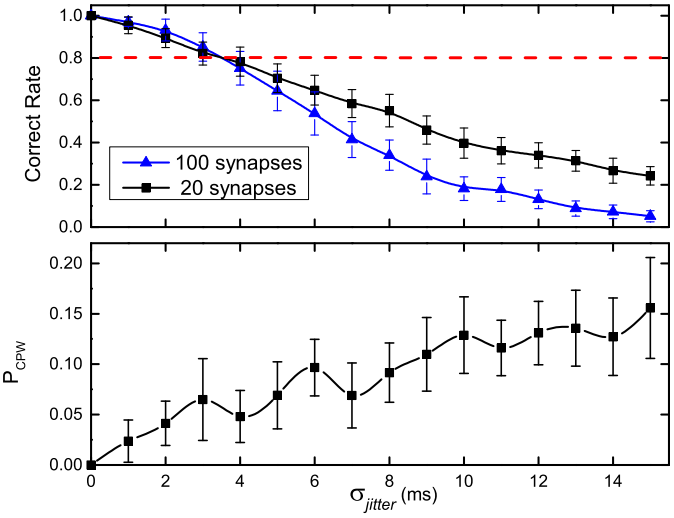

Fig. 7. Recognition rate of learning neurons under the noise-free training mode and the contribution of compete-win samples in the case of 20 afferent neurons.

3.2. Robustness of the ROL rule

In the real world, the capacity for noise tolerance directly affects the performance of a pattern recognition system. For the SNN trained by the ROL rule, classification decisions depend on the first output spike; spiking times are not a concern. In this way, slight deformation of a pattern does not affect the ordering of output spikes, which confirms the high noise tolerance and great generalization ability of the SNN. In this experiment, we investigated the robustness of the ROL rule.

The experiment consists of 10 learning neurons that represent different categories. With $n = 100, 20$ afferent neurons are considered. The encoding time window T_{win} is still 100 ms. First, a set of 10 spike patterns were randomly generated as templates of each category. Each pattern is composed of n spikes that n afferent neurons emit, respectively. There are two training methods for learning neurons, the noise-free training and the noisy training [11]. In the noise-free training, neurons are trained by only 10 template patterns, while 100 patterns with noise were used in noisy training mode. Noisy patterns are generated based on templates by randomly adding jitter to each afferent spike. The jitter obeys a Gaussian distribution with a standard deviation (σ_{jitter}). Up to 100 epochs of training are performed, and the trained SNN is tested

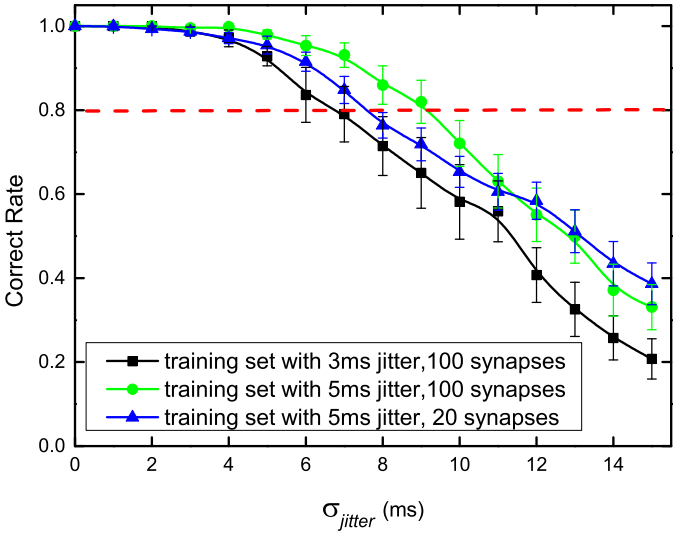

Fig. 8. Recognition rate of learning neurons under the noisy training mode.

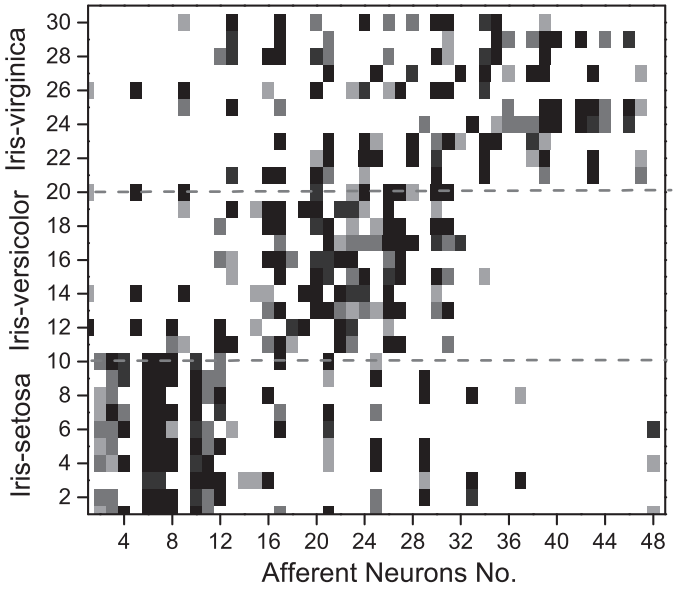

Fig. 9. Spike patterns of Iris dataset samples encoded by Gaussian receptive fields. Darker block means earlier spiking time, while white block means no spike for the corresponding afferent neuron.

by another 200 patterns with noise. Each category has 20 testing patterns.

Fig. 7 shows the results of the experiment in noise-free training mode. With only 10 template training patterns, the weights of neurons quickly converged after several epochs. However, the recognition performance of the test set sharply declines when input jitter is applied. In the case of 100 afferent neurons, the rate of correct performance still reaches approximately 85% with $\sigma_{jitter} = 3$ ms, while only 53% with $\sigma_{jitter} = 6$ ms. Similarly, when the input dimension decreases (20 afferent neurons), the correctness rate decreases almost linearly. The performance of the SNN with 20 afferent neurons is noticeably greater than the performance with 100 afferent neurons, especially when noise strengthens. In general, with more afferent neurons, the efficacy of the SNN should be better. However, the SNN is trained by only 10 template patterns in this experiment, limiting its generalization ability. With more afferent neurons, the similarity between template patterns and noisy patterns decreases when exposed to the same jitter level. In the case of 100 afferent neurons, we observed that nearly all samples that could not be recognized correctly are those that caused learning neurons to remain silent. Conversely, a certain number of samples exist that are able to drive more than one learning neuron to fire and are still classified correctly in the case of 20 afferent neurons. We call these samples compete-win samples, and we calculate the contribution of compete-win samples, P_{CPW}, as follows:

$$P_{CPW} = \frac{N_{CPW}}{N_{RC}} \tag{14}$$

where N_{CPW} is the number of compete-win samples and N_{RC} is the number of samples that are recognized correctly. P_{CPW} is always near 0 in the case of 100 afferent neurons. However, as the bottom of Fig. 7 shows, P_{CPW} in the case of 20 afferent neurons increases as the jitter level strengthens. Because of the improvement in P_{CPW}, the SNN with 20 afferent neurons can maintain a higher correct recognition rate when exposed to a high jitter level. This finding provides evidence that the robustness of the SNN benefits from the core principle of the ROL rule, which states that the first spike generates outcomes. The evidence also suggests that decreasing the dimension of input data can help to improve the robustness of the SNN trained by the ROL rule when the number of training samples is extremely limited.

Fig. 8 shows the results of an experiment in noisy training mode. With sufficient training samples, the robustness of the SNN significantly improves. Compared with a performance rate of 53% for a test set with 6 ms of jitter in noise-free training mode, the rate can reach 95% for the same test set after the SNN is trained by the training set with 5 ms of jitter. Trained by the patterns with higher noise levels, the SNN develops better robustness. Moreover, the comparison between the scenarios of 100 afferent neurons and 20 afferent neurons is different from that in noise-free training mode. For most cases, the SNN with more afferent neurons behaves better in noise training mode. Understandably, the influence of each spike is reduced when the number of afferent neurons increases. With more afferent neurons, the SNN can obtain greater generalization ability if the training set is abundant and representative. However, in the case of very high noise, the SNN with fewer afferent neurons, which is more likely to respond to patterns by emitting spikes, still behaves better. According to the FSMD principle, the SNN with fewer afferent neurons still has the capacity to classify the highly noisy patterns.

3.3. Experiment on the Iris dataset

In the following experiments, we demonstrate pattern recognition ability with the application of the ROL rule using real-world stimuli. First, the linearly non-separable Iris dataset is used in this section. The dataset consists of 150 samples, 50 for each Iris plant category, and each sample has four continuous variables: sepal length, sepal width, petal length, and petal width.

The encoding method is one of the primary problems that restricts the application of the SNN. To encode the continuous feature variables into spikes, a method based on groups of receptive fields is often implemented and fully studied [6]. We also use this method to encode the Iris dataset. For a variable x with a range of $[x_{min}, x_{max}]$, n neurons with different Gaussian receptive fields are used to encode this variable. The ith neuron has the centre $\mu_i = x_{min} + (2i-3)/2 * (x_{max} - x_{min})/(n-2)$ and the width $\sigma_i = 2/3 * (x_{max} - x_{min})/(n-2)$. Then, the variable x is encoded as n activation values computed by the Gaussian receptive fields. Finally, these activation values are translated into spike times in a linear fashion. In particular, $n = 12$ neurons and encoding time window $T_{win} = 50$ ms are considered in this experiment, and the neuron with the activation value below 0.1 does not fire during the simulation phase. Thus, each pattern consists of a maximum of

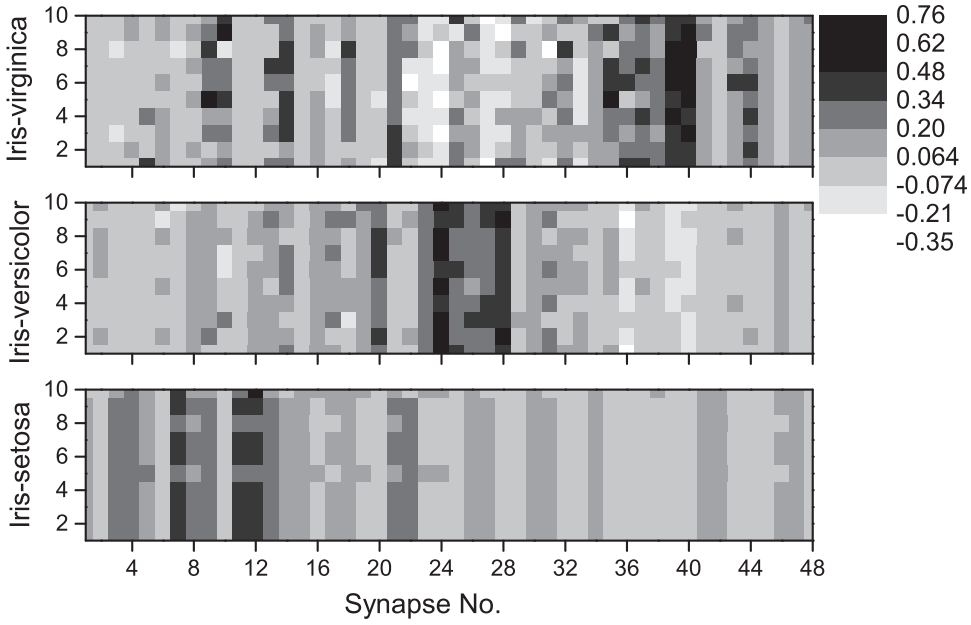

Fig. 10. The synaptic weights of three learning neurons after training in the case of ten-fold cross validation.

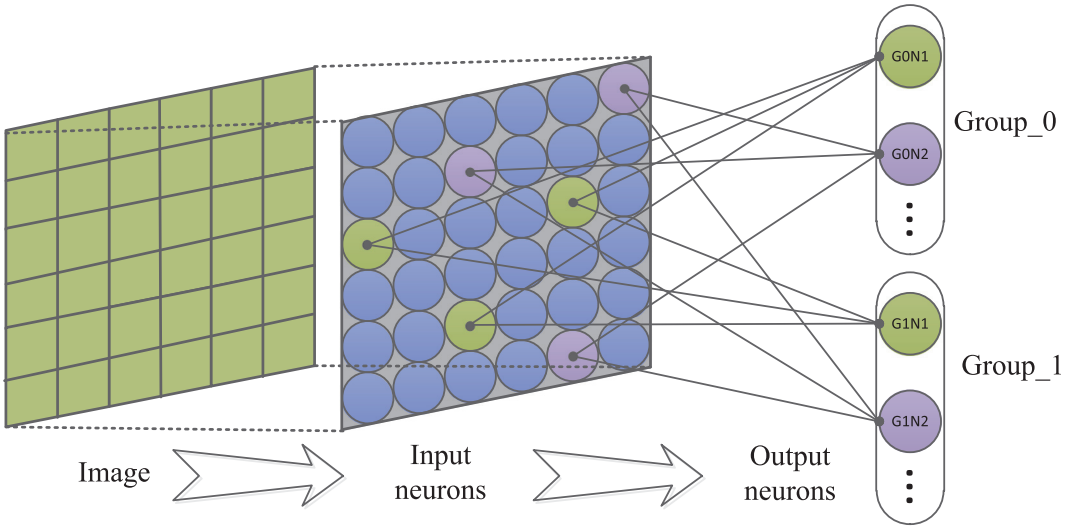

Fig. 11. The architecture of the SNN for handwritten digit recognition.

48 afferent spikes. The results of encoding are shown in Fig. 9 (30 samples). There are only approximately 10 to 13 valid spikes for each pattern, achieving the effect of sparse coding.

The SNN is still the fully connected single-layer network, and each learning neuron associates with one Iris class. Ten-fold cross validation is utilized to demonstrate the generalization ability using the ROL rule. Then, 150 samples are randomly divided into ten groups, and each group contains 15 samples with 5 samples in every Iris plant category. Training is stopped when the number of training epoch reaches 100 or the training accuracy exceeds 97.5%. We perform 100 runs of training and testing. The random distribution of sample groups and the initialization of synaptic weights are repeated in each run.

Fig. 10 shows the synaptic weights distribution of three learning neurons after training in the case of ten-fold cross validation. Considering the encoding result shown in Fig. 9, the outcome strongly demonstrates the adaptability and the availability provided by the ROL rule. The spike patterns of Iris Setosa are very different from the patterns of the other two Iris categories, so the synaptic weights of the learning neuron associated with Iris Setosa converge to relatively stable values in each cross validation. However, large overlapping regions exist among the spike patterns of Iris Versicolor and Iris Virginica, indicating that they are not linearly separable from each other. Trained by the ROL rule, learning neurons automatically discover their differences from one another. The neuron associated with Iris Versicolor achieves negative weights for Synapse Nos. 36–40, and the neuron associated with Iris Virginica achieves negative weights for Synapse Nos. 20–28. The appropriate negative weights ensure neurons' low sensitivity to unassociated patterns but do not affect their first firing to matched patterns. Finally, the average recognition accuracy for the training set is 99.1%, and the average recognition accuracy for test set is 94.2%. Table 1 shows the comparison of recognition accuracy between the ROL rule application and other existing algorithms on the Iris dataset.

From Table 1, we can see that the ROL rule achieves a good balance between the recognition performance and the implementation cost. Compared with the Tempotron [28] and the SpikeProp

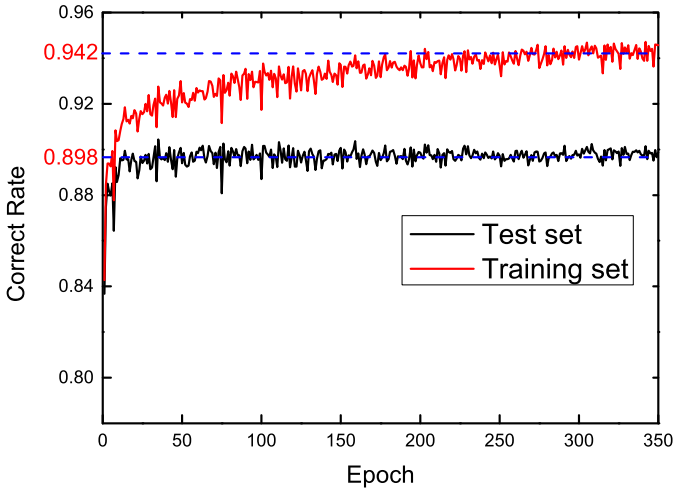

Fig. 12. Learning process of the SNN trained by the ROL rule.

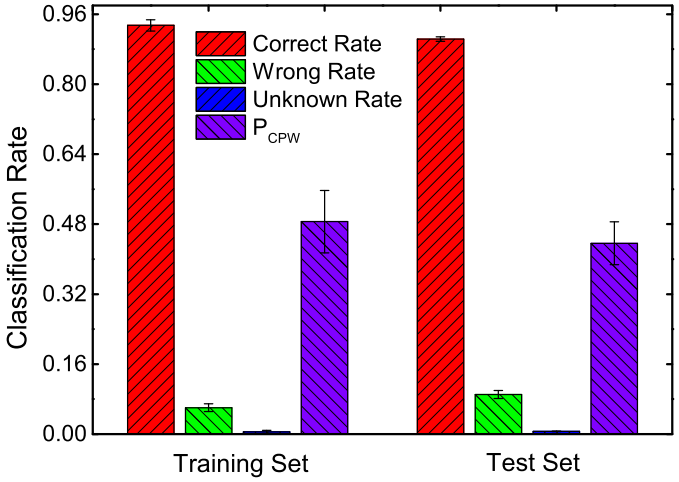

Fig. 13. The classification performance of the ROL rule for the MNIST dataset.

[34], the ROL rule undeniably outperforms in recognition accuracy for the test set and in training cost. The ROL rule achieves comparable accuracy for the test set with fewer training epochs in comparison to other pioneered algorithms such as the GPSNN [33] and the GMSES [7]. In fact, the architecture of the SNN trained by the ROL rule is the simplest and has greater learning efficiency than other algorithms except the ASA [15].

3.4. Experiment on the MNIST dataset

In this section, we use the MNIST dataset to demonstrate the ROL rule. The MNIST dataset contains a large number of grey-scale images of handwritten digits (digits 0–9), 60,000 samples for the training set, and 10,000 samples for the test set. Each image has 28×28 pixels, the value of which ranges from 0 to 255.

To recognize these digital images with the SNN, the means by which to encode pixels into spikes should be considered first. There are many methods that researchers have developed to transform image data to spatiotemporal spike patterns. A rate-based method, in which each pixel is mapped to an encoding neuron with a firing rate proportional to the intensity of the pixel, is used by [36]. In [28], an encoding neuron randomly chooses M pixels to form a binary string based on intensity threshold. Then, the binary string is converted to a decimal value indicative of the firing time of the encoding neuron. Complex pre-process methods are also utilized to improve the performance of the SNN, such as DoG filters [37]. However, these methods increase the complexity and the cost of the recognition system.

To demonstrate the efficacious performance of the ROL rule, we utilize an extremely simple encoding method. Each pixel represents an encoding neuron. The intensity of the pixel is converted into spiking time linearly, with higher intensity mapping to faster spiking time. In this way, most encoding neurons fire near the end of the encoding time window. Fig. 11 shows the entire recognition system of handwritten digits. Considering that information generally is transferred and expressed through populations of neurons, not just one single neuron, population expression is used to enhance the robustness of SNN. This helps to avoid the catastrophic effects caused by the misclassification of a single output neuron. In the output layer, each learning neuron is connected to n encoding neurons that are randomly chosen from the input layer, and m learning neurons constitute a neuron group to code for a digit. Specifically, corresponding neurons in different groups have the same connection with the input layer. We choose $n = 500$, $m = 10$ here.

The decoding of output spikes is different from the SNN with one-to-one relationships between learning neurons and categories. A voting method based on FSMD principle is used to read out the recognition result in this experiment. For Group i of m neurons, T^i_{spike} is the firing time set of this group. The voting weight is calculated using the following formula:

$$W^i_{voting} = \sum_{t \in T^i_{spike}} \left(1 + \frac{T_{win}}{m(T_{win} + t)}\right) \tag{15}$$

$T_{win} = 100$ ms is the encoding time window. According to the formula, the number of output spikes in a neuron group is the major determinant of the final result, while the distribution of spike times is the primary determinant when the spike number is the same for different groups. In this case, the faster spike plays a larger role in the voting weight.

In this experiment, we randomly choose 10,000 examples from the MNIST training set as our training set, 1000 images for each category, and the whole MNIST test set is used to verify the trained SNN. During the training process, image samples from the training set are randomly presented to the SNN and the recognition performance is verified on both the test set and the training set after an epoch of 10,000 samples training.

Table 2 shows the detailed recognition results from a series of experiments. The left-most column is the rate of correct recognition of each digit (CR_D) and the bottom row is the rate of correct response of each neuron group (CR_G). We can see that D2 (digit 2) and D5 are the most difficult to be recognized correctly (83.6%/84.9%), and D8 mixes with a variety of features of other digits, therefore its associated neuron group (G8) obtained a relatively worse rate of correct response (80.6%). The results are essentially consistent with evidence concerning the visual perception of human eyes. Fig. 12 shows the learning process of network. It indicates that using the ROL rule results in rapid convergence ability. The SNN can obtain an adequate rate of correct recognition after dozens of epochs of training. In fact, the highest rate of correct recognition is usually obtained within 50 epochs. As training continues, the correct recognition rate of the training set improves and finally reaches a relatively stable range. In Fig. 12, the average rate of correct recognition for the training set from epoch 250 to 350 is 94.2% and the average rate for the test set is 89.8%. However, the correct recognition rate of the test set declines to some extent in the late stage of training, which means an overfitting phenomenon occurs.

To avoid the overfitting problem, we stop the training process early during the experiment based on consideration of the

current performance and the learning curve presented in Fig. 12. After many experiments with different random initial weights, we calculate the final average performance of the SNN trained by the ROL rule as shown in Fig. 13. The rate of correct recognition is 93.8% for the training set and 90.3% for the test set. The rate of incorrect recognition, which represents the percent of samples that are mistakenly classified, is approximately 6.0% for the training set and approximately 9.1% for the test set. The unknown samples that cause all learning neurons to remain silent are less than 1% of the total number of samples in both the training set and the test set. The P_{CPW} is approximately 45%, which reflects the robustness of the ROL rule for real-world stimuli. Limited by the structure and the size of the network, the recognition performance of the SNN discussed here is not as successful as some SNNs presented by other authors [36,38]. However, as Table 3 shows, these accurately performing SNNs benefit from a large network scale and complex network structure. Additionally, some networks preprocess images for better classification. As a comparison, the SNN trained by the ROL rule is a simple feed-forward network that consists of only 100 learning neurons. This SNN still achieves a relatively comparable and acceptable rate of correct recognition on the test set, which fully embodies remarkable generalization ability when applying the ROL rule. The extremely low cost of implementation and the fast convergence ability are also highlights of the ROL rule.

3.5. Comparison between SNNs and traditional ANNs

Traditional artificial neural networks such as MLPs and CNNs represent the state-of-art for many machine learning and image recognition problems. However, the heavy computational load in traditional ANNs restricts their application in real-time task. On the contrary, the brain-like spike-based computation mode of SNNs improves the energy efficiency and reduces classification latency, which shows the great attraction of SNNs in tasks where fast and efficient recognition plays a crucial role [40,42]. In this section,we take comparative study of these two models to further validate the advantages of SNNs and the ROL rule.

Take MLP network as an example. Considering the SNN with 100 learning neurons in Section 3.4, we build a three-layer MLP network (784-90-10) with ReLU activation function ($a_i^l = max(0, \sum_j w_{ij} a_j^{l-1} + b_i^l)$). The network was trained by the same training set used in Section 3.4 and finally it achieved 99.3% accuracy. The accuracy for testing set is 95.5%. As for computational load, $784 \times 90 + 90 \times 10 = 71,460$ multiplications and $(783 + 1) \times 90 + (89 + 1) \times 10 = 71,460$ adds are required for processing a single image.

To analyse the computational load of the SNN trained by the ROL rule, the membrane potential $V(t)$ is redescribed as:($V_{rest} = 0$)

$$\begin{aligned} V(t) &= \sum_i w_i \sum_{t_i} K(t - t_i) \\ &= V_{nor} \sum_i w_i \sum_{t_i} (e^{-(t-t_i)/\tau_m} - e^{-(t-t_i)/\tau_s}) \\ &= V_{fall}(t) - V_{rise}(t) \end{aligned} \tag{16}$$

where $V_{fall}(t) = V_{nor} \sum_i w_i \sum_{t_i} e^{-(t-t_i)/\tau_m}$, $V_{rise}(t) = V_{nor} \sum_i w_i \sum_{t_i} e^{-(t-t_i)/\tau_s}$. Thus, the discretization forms of $V_{fall}(t)$ and $V_{rise}(t)$ can be described as:

$$V_{fall}(t + \Delta t) = V_{fall}(t) * e^{-\Delta t/\tau_m} \tag{17}$$

$$V_{rise}(t + \Delta t) = V_{rise}(t) * e^{-\Delta t/\tau_s} \tag{18}$$

When an input spike at synapse i arrives at time t, $V_{fall}(t)$ and $V_{rise}(t)$ are updated as:

$$V_{fall}(t) = V_{fall}(t) + V_{nor} w_i \tag{19}$$

$$V_{rise}(t) = V_{rise}(t) + V_{nor} w_i \tag{20}$$

In this way, the state update of a spiking neuron needs two multiplications and $(1 + 2 \times N_t)$ adds in each time step (N_t is the number of spikes that arrive at time t). For the SNN in Section 3.4, totally $2 \times T_{sim}/\Delta t \times 100$ multiplications and $(T_{sim}/\Delta t + 2 \times 500) \times 100$ adds are needed to process a single image of MNIST. T_{sim} is the simulation time and Δt is the precision of simulation.

Table 4 shows the comparison between the SNN trained by the ROL rule and the MLP(784-90-10) in performance and computational load. The SNN is trained with high simulation precision ($\Delta t = 0.1ms$) and quantified on the MNIST test set with lower precision. When Δt increases, we found that the performance of the SNN is almost unaffected, even behaves better with $\Delta t = 1.0$ ms. This further confirms the great robustness of the SNN trained by the ROL rule. On the other hand, the computational load is greatly reduced as the simulation precision decreases. Though the number of additions in the SNN is always higher than that in the MLP, we thought multiplication is the main factor of computational load ([43] shows that the cost of multiplication is dozens times of that of addition). With $\Delta t = 2.0$ ms, only 15% of multiplication operations are performed in the SNN compared with the load of the MLP network.

As can been seen from Table 4, the SNN architecture still can not keep pace with the traditional ANNs in terms of classification accuracy. Considering the single-layer architecture of the SNN, around 90% accuracy is reasonable and acceptable. Not only that, the SNNs have shown great potential as a solution for fast and efficient recognition tasks. Future directions of our research include the ROL rule with monotonically decreasing kernel function, realizing completely event-based computing model to further reduce computational load, and multi-layer architecture to improve the performance of SNN in pattern recognition tasks.

4. Conclusion

In this paper, we have developed a new learning algorithm, the ROL rule, to train spiking neurons to memorize and recognize spatiotemporal spike patterns. Experiments in neuroscience have suggested that the first spike plays a critical role in the information processes of brain. Therefore, we adopt FSMD as the core principle to build the ROL rule upon. According to the relative ordering of spikes that learning neurons emit, the synaptic weights are adapted to make the neuron respond to its associated patterns faster than other neurons. When neurons incorrectly remain silent, the PA process is implemented to push the neurons to fire as soon as possible, and the EPA process drives the neuron to fire earlier to dictate the recognition decision. More importantly, any inappropriately high activity of a neuron is suppressed by the PDHA process, but the process still ensures that the neuron has enough sensitivity to its associated patterns. Without the restriction of specific spiking times as learning targets, the neuron trained by the ROL rule can avoid the contradiction between the afferent patterns and the improper target spiking times. Compared with algorithms that wholly abandon the temporal information of output spikes, such as the Tempotron rule [29], the ROL rule has a greater capacity for the characterization of complex real-world stimuli.

The ROL rule is verified using the Iris dataset and the MNIST dataset. The results have highlighted the remarkable features of the ROL rule: great robustness, generalization ability, and fast convergence. However, limitations in the ROL rule exist. A multi-layer SNN cannot be trained by the ROL rule directly at the present time. Without high-level abstraction ability formed by a multi-layer network, it is difficult to improve the SNN to deal with more complex pattern recognition tasks. For our future research, developing a learning algorithm based on the ROL rule for a multi-layer SNN is one of our main objectives.

References

[1] W. Maass, Networks of spiking neurons: the third generation of neural network models, Neural Netw. 10 (9) (1997) 1659–1671.
[2] V.J. Uzzell, E.J. Chichilnisky, Precision of spike trains in primate retinal ganglion cells, J. Neurophysiol. 92 (2) (2004) 780–789.
[3] Z.F. Mainen, T.J. Sejnowski, Reliability of spike timing in neocortical neurons, Science 268 (5216) (1995) 1503–1506.
[4] W. Bair, C. Koch, Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey, Neural Comput. 8 (1996) 1185–1202.
[5] A. Borst, F.E. Theunissen, Information theory and neural coding, Nature Neurosci. 2 (11) (1999) 947–957.
[6] S.M. Bohte, J.N. Kok, H.L. Poutré, Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing 48 (2002) 17–37.
[7] Y. Xu, X. Zeng, L. Han, J. Yang, A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks, Neural Netw. 43 (2013) 99–113.
[8] T. Masquelier, R. Guyonneau, Competitive STDP-based spike pattern learning, Neural Comput. 21 (5) (2009) 1259–1276.
[9] F. Ponulak, A.J. Kasinski, Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting, Neural Comput. 22 (2010) 467–510.
[10] A. Mohemmed, S. Schliebs, SPAN: spike pattern association neuron for learning spatio-temporal spike patterns, Int. J. Neural Syst. 22 (4) (2012) 1–17.
[11] Q. Yu, H. Tang, Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns, PLoS One 8 (11) (2013) :e78318.
[12] R.V. Florian, The chronotron: a neuron that learns to fire temporally precise spike patterns, PLoS One 7 (8) (2012) :e40233.
[13] Q. Yu, R. Yan, H. Tang, A spiking neural network system for robust sequence recognition, IEEE Trans. Neural Netw. Learn. Syst. 27 (3) (2016) 621–635.
[14] J. Hu, H. Tang, H. Li, L. Shi, A spike-timing-based integrated model for pattern recognition, Neural Comput. 25 (2013) 450-472.
[15] X.R. Xie, H. Qu, Z. Yi, Efficient training of supervised spiking neural network via accurate synaptic-efficiency adjustment method, IEEE Trans. Neural Netw. Learn. Syst. PP (2016) 1–14.
[16] S. Thorpe, J. Gautrais, Rank order coding, Comput. Neurosci. 13 (1998) 113–119.
[17] S. Thorpe, A. Delorme, Spike based strategies for rapid processing, Neural Netw. 14 (6) (2001) 715-726.
[18] S. Wysoski, L. Benuskova, N. Kasabov, Evolving spiking neural networks for audiovisual information processing, Neural Netw. 23 (7) (2010) 819–835.
[19] K. Dhoble, N. Nuntalid, Online spatio-temporal pattern recognition with evolving spiking neural networks utilising address event representation, rank order, and temporal spike learning, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, 2012, pp. 1–7. June 10–15
[20] G. Tim, M. Markus, Rapid neural coding in the retina with relative spike latencies, Science 319 (5866) (2008) 1108–1111.
[21] G. Portelli, J.M. Barrett, Rank order coding: a retinal information decoding strategy revealed by large-scale multielectrode array retinal recordings, Eneuro 3 (3) (2016) 1–18.
[22] S.J. Johansson, I. Birznieks, First spikes in ensembles of human tactile afferents code complex spatial fingertip events, Nat. Neurosci. 7 (2004) 170–177.
[23] C.P. Hung, G. Kreiman, Fast readout of object identity from macaque inferior temporal cortex, Science 310 (5749) (2005) 863–866.
[24] H.T. Chen, K.T. Ng, A. Bermak, M.K. Law, Spike latency coding in a biologically inspired microelectronic nose, IEEE Trans. Biomed. Circuits Syst. 5 (2011) 160–168.
[25] E.K. Kim, G.J. Gerling, S.A. Wellnitz, Using force sensors and neural models to encode tactile stimuli as spike-based response, in: Proceedings of the IEEE Haptics Symposium, HAPTICS 2010, Waltham, MA, USA, 2010, pp. 195–198. March 25–26
[26] T. Masquelier, S.J. Thorpe, Unsupervised learning of visual features through spike timing dependent plasticity, Plos Comput. Biol. 3 (2) (2007) 247–257.
[27] H. Mostafa, Supervised learning based on temporal coding in spiking neural networks, CoRR (2016) 1-9. abs/1606.08165
[28] Q. Yu, H. Tang, K.C. Tan, A brain-inspired spiking neural network model with temporal encoding and learning, Neurocomputing 138 (2014) 3–13.
[29] R. Gütig, H. Sompolinsky, The tempotron: a neuron that learns spike timing-based decisions, Nat. Neurosci. 9 (3) (2006) 420–428.
[30] A. Hodgkin, A. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. 117 (4) (1952) 500–544.
[31] E.M. Izhikevich, Resonate-and-fire neurons, Neural Netw. 14 (2001) 883–894.
[32] E.M. Izhikevich, Simple model of spiking neurons, IEEE Trans. Neural Netw. 14 (6) (2003) 1569–1572.
[33] S. Dora, S. Suresh, N. Sundararajan, A two stage learning algorithm for a growing-pruning spiking neural network for pattern classification problems, in: Proceedings of the International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, 2015, pp. 1–7. July 12–17
[34] S. Ghosh-Dastidar, H. Adeli, Improved spiking neural networks for eeg classification and epilepsy and seizure detection, Integr. Comput. Aided Eng. 14 (3) (2007) 187–212.
[35] J.J. Wade, L.J. McDaid, J.A. Santos, SWAT: a spiking neural network training algorithm for classification problems, IEEE Trans. Neural Netw. 21 (11) (2010) 1817–1830.
[36] P.U. Diehl, M. Cook, Unsupervised learning of digit recognition using spike-timing-dependent plasticity, Front. Comput. Neurosci. 9 (99) (2015) 99, doi:10.3389/fncom.2015.00099.
[37] Q. Yu, H. Tang, H. Li, Rapid feedforward computation by temporal encoding and learning with spiking neurons, IEEE Trans. Neural Netw. Learn. Syst. 24 (10) (2013) 1539–1552.
[38] P.U. Diehl, D. Neil, J. Binas, Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing, in: Proceedings of the International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, 2015, pp. 1–8. July 12–17
[39] S. Hussain, S.-C. Liu, A. Basu, Improved margin multi-class classification using dendritic neurons with morphological learning, in: Proceedings of the IEEE International Symposium on Circuits and Systemss, ISCAS 2014, Melbourne, Victoria, Australia, 2014, pp. 2640–2643. June 1–5
[40] P. O'Connor, D. Neil, T. Delbruck, Real-time classification and sensor fusion with a spiking deep belief network, Front. Neurosci. 7 (2013) 178.
[41] M. Beyeler, N.D. Dutt, J.L. Krichmar, Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule, Neural Netw. 48 (2013) 109–124.
[42] Y. Cao, Y. Chen, D. Khosla, Spiking deep convolutional neural networks for energy-efficient object recognition, Int. J. Comput. Vis. 113 (1) (2015) 54–66.
[43] S. Han, X. Liu, H. Mao, J. Pu, EIE: efficient inference engine on compressed deep neural network, ACM Sigarch Comput. Archit. News 44 (3) (2016) 243–254.

Zhitao Lin received his B.S. degree in electronic and information engineering from College of Electrical Engineering, Zhejiang University, China, in 2013. He is a Ph.D. student in Institute of VLSI design, Zhejiang University, China at present. His research interests include System-on-Chip (SoC) design, deep learning accelerator and learning algorithms for spiking neural networks.

De Ma was born in Jun. 20th, 1985. He received the B.S degree in electrical and information engineering from Zhejiang University, Hangzhou, China, in 2004. Then he pursued his Ph.D. study in the Institute of VLSI Design, Zhejiang University. From Oct. 2010 to Jan. 2011, he was an intern at VERIMAG Lab. France. He is now the associate professor in the Department of Electronic and Information, Hangzhou Dianzi University. His research interests include Neuromorphic, SoC Architecture exploration and MPSoC performance estimation, etc.

Jianyi Meng received his B.S. degree and Ph.D degree in Department of Information Science & Electronic Engineering, Zhejiang University, 2004 and 2009. He joined Institute of VLSI Design, College of Electrical Engineering, Zhejiang University in 2009 as a postdoctor. He visited IMEC, Belgium, as short term scholar visitor in 2010. He is now a professor of State Key Laboratory of ASIC and System, Department of Microelectronics, Fudan University. He has been in charge of or has participated as a main contributor in Special Grand National Science-technology Project, Pre-research Program, 863 Program, and etc. He has published over 30 academic papers index by SCI/EI journals, and applied over 20 patents. His research mainly focuses on computer architecture, ultra-low power digital circuit design and application specific SoC design.

Linna Chen received her B.S. degree in electronic and information engineering from Tianjin University, China, in 2015. She is a M.S. student in Institute of VLSI design, Zhejiang University, China at present. Her research work focuses on SoC verification, deep learning accelerator, and learning algorithms in spiking neural networks.