Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A novel spiking neural network of receptive field encoding with groups of neurons decision*

Yong-qiang MA†, Zi-ru WANG, Si-yu YU, Ba-dong CHEN, Nan-ning ZHENG†, Peng-ju REN†‡

Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China

E-mail: musaqiang@stu.xjtu.edu.cn; nnzheng@mail.xjtu.edu.cn; pengjuren@mail.xjtu.edu.cn

Received Oct. 31, 2017; Revision accepted Jan. 22, 2018; Crosschecked Jan. 25, 2018

Abstract: Human information processing depends mainly on billions of neurons which constitute a complex neural network, and the information is transmitted in the form of neural spikes. In this paper, we propose a spiking neural network (SNN), named MD-SNN, with three key features: (1) using receptive field to encode spike trains from images; (2) randomly selecting partial spikes as inputs for each neuron to approach the absolute refractory period of the neuron; (3) using groups of neurons to make decisions. We test MD-SNN on the MNIST data set of handwritten digits, and results demonstrate that: (1) Different sizes of receptive fields influence classification results significantly. (2) Considering the neuronal refractory period in the SNN model, increasing the number of neurons in the learning layer could greatly reduce the training time, effectively reduce the probability of over-fitting, and improve the accuracy by 8.77%. (3) Compared with other SNN methods, MD-SNN achieves a better classification; compared with the convolution neural network, MD-SNN maintains flip and rotation invariance (the accuracy can remain at 90.44% on the test set), and it is more suitable for small sample learning (the accuracy can reach 80.15% for 1000 training samples, which is 7.8 times that of CNN).

Key words: Tempotron; Receptive field; Difference of Gaussian (DoG); Flip invariance; Rotation invariance
https://doi.org/10.1631/FITEE.1700714 **CLC number:** TP183

1 Introduction

The deep neural network (DNN) draws on the hierarchical structural features of the biological neural network (Hannun et al., 2014; LeCun et al., 2015; Schmidhuber, 2015). Parameters of the model are adjusted by training, and a group of better parameters for a specific model is ultimately obtained. Although this kind of model can do classification well, the construction of the network is difficult, and the training cost is high (He et al., 2016).

The spiking neural network (SNN) is considered as the third-generation neural network (Maass, 1997). Its main characteristic is using the spike as the basic element for information transmission, different from DNN which is based on numerical calculation. There are several important properties of SNN, and the receptive field is the first. The receptive field is a particular region of an individual sensory neuron, in which a stimulus will modify the firing of that neuron. Receptive field profiles of different types are found in biological vision, and these are tuned to be sensitive to sizes and orientations for image perception, and visual systems should be invariant to the natural types of image transformations that occur in their environment (Gilbert and Wiesel, 1992; Coates et al., 2011).

Spike is also called an action potential, and it is generated by the flow of charged particles in

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61773312, 61773307, and L1522023), the China Postdoctoral Science Foundation (No. 2016M590949), and the National Basic Research Program (973) of China (No. 2015CB351703)

ORCID: Peng-ju REN, http://orcid.org/0000-0003-1163-2014

biological neurons. These are discrete events that take place at points in time, rather than continuous values (Thorpe et al., 2001). When the external signal produces a stimulus, the signal will be encoded in the form of spike trains and transmitted inside the biological neural network. The neuron receives the spike signals from others through the synapses and accumulates different potentials according to different strengths of the connections. This will raise the membrane potential of the neuron. When the accumulated potential reaches a certain threshold, the neuron will fire a spike through the axon to release potential and then the spike will pass through the synapse to reach other neurons. The fired neuron comes into an absolute refractory period, during which the potential of the neuron will sharply drop to a low value and the neuron becomes responseless to input spikes, until its potential recovers to a stable value.

The neurons have different initial weights of synapses when the network is built. Different synaptic weights will generate different models. Not every neuron can give the correct output in all models. Thus, using more groups of neurons can improve the accuracy of the model (Yu et al., 2014). An MD-SNN model creates more spike trains by one image, which is more suitable for more groups of neurons to make a decision. A similar structure of SNN has already been used for fMRI data, leading to a satisfactory result (Ma et al., 2017).

These properties are crucial for biological neural networks and need to be considered altogether. However, researchers always overlook one or many of them. As far as we know, this is the first study in which the receptive field, the refractory period of the neuron, and group-neuron decision are considered simultaneously in constructing the SNN model. Specifically, in our SNN model, the images are encoded to the spike trains by a receptive field model in view of signal processing and a biological vision nerve model, respectively. These spike trains will be inputted into the biological neurons for analysis. Then we apply a random selection strategy for spikes to simulate the absolute refractory period (Yeomans, 1979; Berry and Meister, 1998) of the real neuron. Because the neuron does not receive the spikes during the absolute refractory period, an individual neuron does not have to handle all the information. The proposed SNN model also uses numbers of neurons to deal with information jointly. The strategy is using multiple groups of neurons to make a common decision, and each neuron just receives part of the selected contents of the encoded images. This strategy further ensures that, even if each neuron receives only partial input of spikes, all the information of the image is still fully used. This also reduces the burden of each neuron in processing information in the hidden layer.

2 Spiking neural model

SNN is a more biologically relevant neuromorphic model, and it conveys information through spike trains rather than numerical values (Ghosh Dastidar and Adeli, 2007). The neural model used in the traditional neural network is based on numerical calculation and is implemented using the computational platform of von Neumann architecture. The advantages of numerical calculation depend on its high efficiency and accuracy. The biological neuron does not distinguish between the operation and storage very clearly, so the information transmission and storage are closely related with synaptic connection changes. The brain works in an event-driven way, and it can save more resources than synchronous clock operations.

In recent years, different types of bionic spiking neural models have been proposed (Masquelier et al., 2009; Ponulak and Kasiński, 2010; Wade et al., 2010), such as the Hodgkin-Huxley model (Hodgkin and Huxley, 1952) and the later developed integrate-and-fire model (IF) (Brette and Gerstner, 2005), resonate-and-fire model (Izhikevich, 2001), Izhikevich (IM) model (Izhikevich, 2003), and other spiking neural models. The calculation equation for the IF model is simpler and easier to solve (Izhikevich, 2004). Therefore, in this study, we select the IF model as the neural model to simulate the spiking distribution of neurons to analyze the performance of our constructed network structure. The simplest dynamic equation of the IF model can be described by the following expression:

$$\tau_{\mathrm{m}} \frac{\mathrm{d}v}{\mathrm{d}t} = -v(t) + RI(t), \tag{1}$$

where τ_{m} represents the constant membrane time, which depends on the membrane resistance and capacitance of the neuron, $v(t)$ represents the membrane potential of the neuron at time t, R is the membrane resistance of the neuron, and $I(t)$ represents the amount of current that is transmitted to the neuron by synapses at time t. When $v(t)$ reaches a definite neural firing threshold V_{th}, the neuron will release the voltage by firing a spike, and the spike will be delivered to other neurons through the synapses. The membrane potential of the neuron which fires the spike is rapidly reduced to a lower value V_{rest}. Here, V_{rest} is called the residual potential. After firing the spike, the membrane potential of the neuron needs a short period to recover from the low value V_{rest} to a steady state. During this time, the neuron will no longer respond to any other input spike or fire spike. This phenomenon is called the absolute refractory period and is also present in biological neurons.

2.1 Tempotron neural model

The Tempotron neural model (Gütig and Sompolinsky, 2006), a typical IF neural model, has already been used in some SNNs and proved effective in image classification. The network structure established in this study will be applied to the image classification problem eventually, where the membrane potential of a neuron is determined by the input potential of all presynaptic potentials connected to it, and its synaptic weights are calculated as follows:

$$V(t) = \sum_i \omega_i \sum_{t_i} K(t - t_i) + V_{\mathrm{rest}}, \tag{2}$$

where i is the input spike from the i^{th} synaptic connection, ω_i and t_i represent the weight of the synaptic connection and the time of inputting the spike, respectively, t is the current time, the calculation of the membrane potential of the neuron at the present time is considered only when $t_i < t$, V_{rest} indicates the remaining membrane potential of the neuron at the present time, K represents a normalized postsynaptic potential (PSP) calculation of the kernel function, and we have

$$K(t - t_i) = V_0(\exp(\frac{-(t - t_i)}{\tau_{\mathrm{m}}}) - \exp(\frac{-(t - t_i)}{\tau_{\mathrm{s}}})), \tag{3}$$

where τ_{m} and τ_{s} represent the decay time constants of the membrane integration current and the synaptic current, respectively. There is a fixed proportion of the relationship between τ_{m} and τ_{s}. In general, $\tau_{\mathrm{m}} = 4\tau_{\mathrm{s}}$, and we choose $\tau_{\mathrm{m}} = 20$ ms for all the experiments. V_0 is the normalized constant of the kernel function. This ensures that the peak amplitudes of all the spikes through the synaptic inputs are consistent.

According to the calculation formula of the membrane potential kernel function of the Tempotron neuron model, the variation tendency of a spike can be described as the potential intensity rising from V_{rest} to V_{th} and then gradually decaying and tending to V_{rest} after the generation of a spike. The spike will be fired when the potential of the neuron reaches V_{th}. However, the transmission of the spikes relates not only to the time when it was generated, but also to the connection strength of the synapse, and the different synaptic weights make the incoming spikes express different meanings. When the synaptic weight ω_i is positive, the presynaptic potential produces a long-term potentiation (LTP); when ω_i is negative, the presynaptic potential exerts a long-term depression (LTD) effect (Bi and Poo, 2001; Legenstein et al., 2006). The final postsynaptic potential (PSP) will be determined by the combined effect of LTP and LTD on all synaptic membranes attached to it.

2.2 Learning rule

Receiving and generating spikes are the basic functions of the neuron model, and the change of interconnected synaptic weights between neurons is a key factor in determining whether the network can work effectively. The synaptic weights are determined by the structure of the network, the characteristics of the neurons themselves, and the results of the desired output. The network model established in this study adopts a clear hierarchical structure. The definition of the Tempotron neuron model limits the maximum number of spikes to be fired. The Tempotron neuron can fire just once only during the time window (Gerstner and Kistler, 2002), and will remain silent after the potential is released. We will use the structured network to do image classification on the MNIST data set. So, we define the learning rules of the network according to the expected output classification results combined with the synaptic weight adjustment rules of the Tempotron neuron.

The output spike of the Tempotron neuron model fires when the membrane potential reaches the threshold, and thus a spike is generated and transmitted to the next layer of the neuron through the synaptic connection, and the potential of the

Tempotron neuron drops rapidly and remains low. Then the neuron does not respond to an input signal from the presynaptic membrane during the time window. It will start to work again when the next time window comes. In other words, the Tempotron neuron model has two different output spike trains, which are one spike only and no spike, in the spike train. These two different output states can be defined as two different classification results, so each Tempotron model can be a binary classifier. The synaptic weights of the Tempotron neurons in the network are adjusted according to the comparison between its actual output and the expected output.

If the actual output is consistent with the expected output, the output is correct, and the synaptic weights do not need to be adjusted. We mark this state as R (right). If the actual output and the expected output are inconsistent, the output is wrong, and the synaptic weight needs to be adjusted. We mark this state as W (wrong). According to the learning rule of the Temptron neuron, we can use the following formula to modify the synaptic weight:

$$\Delta\omega_i = \begin{cases} \lambda \sum_{t_i<t_{\max}} K(t_{\max}-t_i), & \text{W+ error}, \\ -\lambda \sum_{t_i<t_{\max}} K(t_{\max}-t_i), & \text{W}- \text{ error}, \\ 0, & \text{R}, \end{cases} \tag{4}$$

where λ is the learning rate, $t_{\max}$ represents the time at which the neural membrane potential is the highest, t_i represents the i^{th} input spike, and K is the calculation of the kernel function of PSP.

There are two cases that can lead to the emergence of the W state (Yu et al., 2014). One is the actual output without a firing spike when it should fire, indicating that the input elements of the synaptic weights are too small to cause the potential accumulation to be sufficient for firing the spike, so the network needs to find the time of the neuron reaching the maximum potential. This is most likely to be the fire spike position, and the network needs to increase the synaptic weights to increase the potential. We define the case of modifying the weights as W+ modification. The other case is defined as W− modification when the actual output includes an unexpected spike, which means that the corresponding synaptic weights need to be reduced to keep silence.

3 Multiple-neuron decision spiking neural network

Generally, neural networks can be divided into three stages: encoding, learning, and readout (Yu et al., 2014). The encoding stage refers to the external stimulus signal through a certain encoding mode. Results of the encoding mode will be transformed into a network, which can identify and accurately handle the results. The learning stage is the core of the network structure. This stage requires a number of training adjustments to make the parameters adapt to the requirements of the problem. This is required to obtain the expected output. The readout stage is aimed to translate the results obtained at the learning stage to information that people can understand. For example, the learning stage gives the probability distribution of each classification, and then the readout stage gives a certain classification according to the probability distribution.

The network structure proposed in this study is also divided into these three stages, each of which is described in detail as follows.

3.1 Encoding

In the encoding stage the input image needs to be transferred into spike trains. Here the encoding method we use is referenced from the human visual sense of the concept of the receptive field to build the model. The visual receptive field refers to the electrical response of a single nerve fiber or a single neuronal cell at a higher level of the visual pathway when a particular region of the retina is stimulated by light. This area is the receptive field of the nerve fiber or cell. The receptive field of human retinal ganglion cells is in a spatial distribution of concentric forms. Rodieck (1965) proposed a mathematical model of the receptive field with antagonistic concentric circles by difference of Gaussian (DoG), which is a central mechanism of excitability and a weaker but larger inhibitory effect peripheral mechanism. These two mechanisms of mutual antagonism both have the properties of Gaussian distribution and are opposite to each other, and the central mechanism has higher peak sensitivity. The DoG model has been widely used in the computer vision field and also in encoding spike trains (Yu et al., 2013). The mathematical expression of the model can be defined as

$$\text{DoG}(\overline{x}) = a_{\text{c}}G(\overline{x};\delta_{\text{c}}) - a_{\text{s}}G(\overline{x};\delta_{\text{s}}), \tag{5}$$

where G is the two-dimensional Gaussian operator at position $\overline{x}(x_1, x_2)$, and the expression is

$$G(\overline{x}; \delta) = \frac{1}{2\pi\delta^2}\exp(-\frac{|\overline{x}|^2}{2\delta^2}) = \frac{1}{2\pi\delta^2}\exp(-\frac{x_1^2 + x_2^2}{2\delta^2}), \tag{6}$$

where a_c and a_s represent the central and surrounding sensitive coefficients, respectively, and δ_c and δ_s represent the standard deviation of the central and surrounding Gaussian operators, respectively.

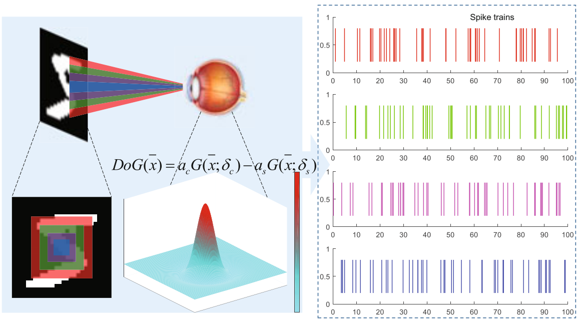
$DoG(\overline{x}) = a_c G(\overline{x}; \delta_c) - a_s G(\overline{x}; \delta_s)$
Spike trains
1
0.5
0
0
10
20
30
40
50
60
70
80
90
100

Fig. 1 Receptive field model of DoG in encoding
When choosing different sizes of vision windows, different spike trains will be generated

For different sizes of images, the DoG operator needs to choose different sizes of the $n \times n$ convolution windows to calculate the weight of each position, and use the $n \times n$ window to convolute the image with padding. Convolution is a biological method for encoding images (Hubel and Wiesel, 1962; Fukushima, 1980). Notice that there exist various operators, such as the Laplacian of Gaussian (LOG) operator (Burt and Adelson, 1983), Sobel operator (Sobel, 2014), Prewitt operator (Al-Amri et al., 2010), and Canny operator (Canny, 1986), some of which are sensitive to orientation, intersection, etc. Table 1 shows that the DoG operator is the most suitable for encoding the MNIST images using our SNN model.

Table 1 Different operators for encoding in our SNN model for MNIST dataset classification

Operator	Accuracy
LoG	85.13%
Sobel	83.68%
Prewitt	83.19%
Canny	83.65%
DoG	90.44%

From the calculation, the time at which the spike is generated on the time axis can be determined. The higher value of the calculation result generates the spike in the forward of the time axis, which means that the response intensity is higher for the stronger stimulus. The lower value will generate the spike in the backward. As shown in Fig. 1, for the same MNIST image, different sizes of vision windows will encode different sequences of spike trains. Each image corresponds to a spike train with a time length of t_window; that is, the time window length during image encoding is fixed at t_window. The spike train (ST) of an image is generated as

$$\mathrm{ST} = \bigcup_{i=1}^{t_\mathrm{window}} \delta(t_i), \tag{7}$$

where $\delta(t_i)$ is the impulse function, and t_i is any time in the time window. The proposed encoding method for the MNIST data set can be illustrated by Fig. 2.

3.2 Learning

The learning stage is the core of the entire network. If the size of a picture is $l \times m$, the convolution encoded with padding by a DoG operator of an $n \times n$ vision window will obtain a spike train with $l \times m$ spikes on a fixed time window t_window. The training set will be used to adjust the synaptic weights in the network until the output accuracy converges to a stable result.

In this study, the network structure for the learning stage adopts the scheme of multiple groups of neurons to make a decision, which we call a multiple-neuron decision spiking neural network (MD-SNN). The analysis of the characteristics of the spike trains is more comprehensive, and the accuracy of the classification is improved.

The concrete structure can be described as follows:

Choosing r groups of spikes by randomly selecting spikes from the whole spike train set to make up r spike trains. From experiments we find that the r should not be greater than 20. Because more groups of neurons have been used, more computing time is needed, and the results have no significant change when r is larger than 20 (Dasgupta et al., 2017; Ma et al., 2017). Each spike train contains some of the spikes; that is, $p \times 100\%$ $(0 < p \leq 1)$ of the spikes are randomly selected from the original spike train to form a new spike train. The selected spikes are discrete. The sub-spike trains (ST_sub) are generated by dot product between the original spike train and

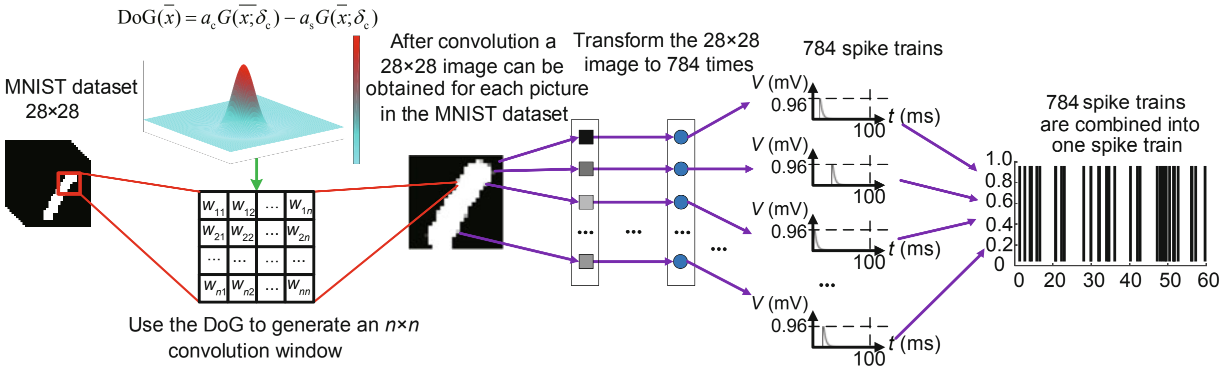
$\text{DoG}(\bar{x}) = a_c G(\bar{x};\delta_c) - a_s G(\bar{x};\delta_c)$
MNIST dataset 28×28
w_{11} w_{12} … w_{1n}
w_{21} w_{22} … w_{2n}
… … … …
w_{n1} w_{n2} … w_{nn}
Use the DoG to generate an n×n convolution window
After convolution a 28×28 image can be obtained for each picture in the MNIST dataset
Transform the 28×28 image to 784 times
784 spike trains
V (mV)
0.96
t (ms)
100
784 spike trains are combined into one spike train
1.0
0.8
0.6
0.4
0.2
0
0 20 30 40 50 60

Fig. 2 The encoding process of MNIST with a receptive field of DoG

A 28 × 28 picture of MNIST is convoluted with padding by an $n \times n$ vision window generated by the DoG model, and then the pixels of the convolution image are mapped to a fixed time window to compose a spike train including 28 × 28 spikes

a random binary vector, with $p = 1$; i.e.,

$$\text{ST}_{\text{sub}} = \text{ST} \cdot \boldsymbol{v}, \tag{8}$$

where $\boldsymbol{v}$ is the random binary vector. The spikes come from the convolution by an $n \times n$ vision window, which is a DoG receptive field, so each spike can describe part of the image. Even if p is very small, patterns would not be changed significantly. The smaller p is, the less computing time is needed. The experimental results show that the complete spike train is basically equivalent to the spike train containing only part of the spikes after Tempotron training, or even part of spikes can train a better model for some value of p. It means that much information in the spike train is redundant for the Tempotron model and is not used efficiently. Therefore, we use a random selection method to generate multiple sets of spike trains. More spike trains can be used for training to obtain more features. Here we use $p \times 784$ spikes of the whole spike train for an MNIST image, since an MNIST image can be encoded to a spike train that includes 784 spikes.

The r groups of spike trains are used as inputs to be transmitted into $r \times s$ Tempotron neurons, where s represents the number of categories; i.e., there are r groups and each group includes s Tempotron neurons. One group of the neurons receives one type of randomly selected spike train from the encoding stage. For each category, there are r Tempotron neurons being used to make a decision. When the neuron fires a spike, it means that the input image belongs to the category of this neuron. No spike firing means that the input image does not belong to this class. In the training phase, the output needs to be compared with the actual type: if they are consistent, the synaptic weights do not need to be modified; if they are not consistent, the network needs to adjust the synaptic weights following the rules. These were described in the previous section.

The outputs of the $r \times s$ Tempotron neurons correspond to the s classification results. Each class includes r outputs for each image. The r results of outputs are input into s neurons, which are used for counting the number of spikes which are fired from the Tempotron neurons in each category. In some other SNN structures multiple neurons are used to make a decision. However, most methods use more than one neuron to make one decision (Dora et al., 2016, 2017). In the proposed MD-SNN, each neuron can give a definite result. In the readout stage all these results are used to compute the probability of each category for an image. The image will belong to the highest probability category.

Fig. 3 illustrates the groups of neurons in the learning phase of MD-SNN on the MNIST dataset. There are r input neurons, which send sub-spike trains to $r \times s$ Tempotron neurons, with s representing the number of categories.

3.3 Readout

The readout stage gives the final classification results that depend on the statistical results from the learning stage. We define that the image belongs to the category in which the number of fired Tempotron neurons is the largest. When there is only one category which has the largest number of fired Tempotron neurons, the input picture belongs to this

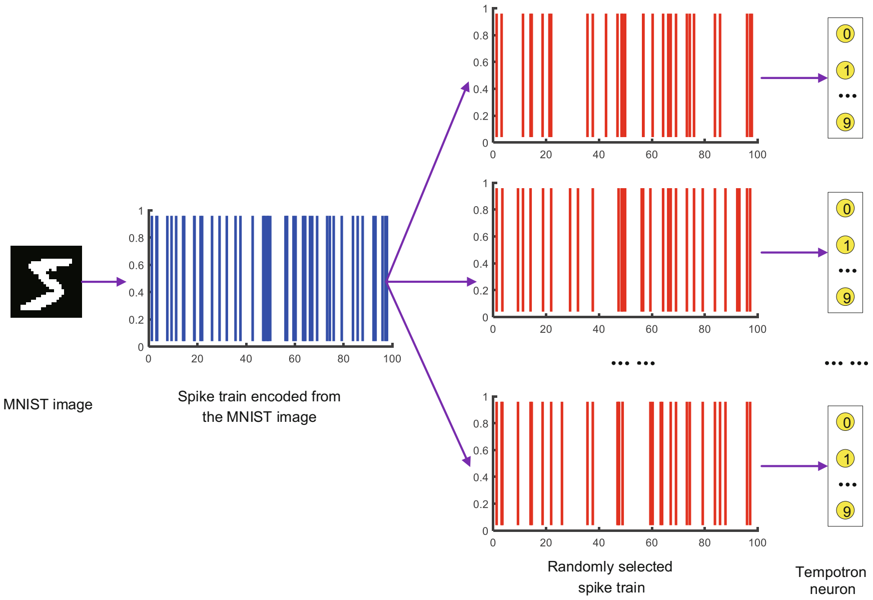
MNIST image
Spike train encoded from the MNIST image
Randomly selected spike train
Tempotron neuron
0
1
9

Fig. 3 A group of neurons in the learning phase of MD-SNN on the MNIST dataset
The images of MNIST are first encoded into spike trains (in blue), and then several sub-spike trains (in red) are generated by randomly selecting ratio p from the original spike trains. The randomly selected spike trains will be inputted into 10 Tempotron neurons, which represent 10 classes of handwritten digits

category; if the same number of fired Tempotron neurons are outputted, the input picture is considered to belong to the first category. The readout stage is shown in Fig. 4.

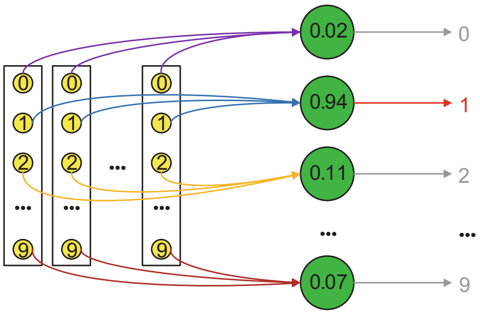

Fig. 4 The readout stage at which the outputs of multiple groups of neurons are counted
The classification result is given based on the statistic result. The input image belongs to the class where the neuron has the largest count number

4 Performance of MD-SNN

We tested the proposed MD-SNN structure on the handwritten digital MNIST dataset and obtained 90.44% accuracy in classification.

The MNIST dataset contains two parts, the training set and the test set. The training set contains 60 000 handwritten pictures. There are 10 000 pictures in the test set. Each picture is a grayscale image with 28×28 pixels and represents a handwritten digit from 0 to 9, and the probability of occurrence of each digit is the same. In our method, first, the images are encoded by the $n \times n$ ($n = 1, 3, 5, 7, 9$) vision window, and the weights in the vision window are calculated by the DoG operator. After the convolution with padding, we can obtain the encoding results with the same size of original images. Normalizing each picture, a spike train that contains 784 spikes corresponding to a time window of 100 ms will be generated. Twenty randomly selected spike trains, which include the sp (sp $= 784 \times p$) spikes, are created by the generated spike train. These randomly selected spike trains are the inputs for Tempotron neurons. Here we choose p from 0.1 to 1 and the step size is 0.1. There are 10 digits that need to be divided into 10 classes, and 20 groups of spike trains are generated by a random sequence of selected spikes. Therefore, 200 Tempotron neurons are required to form an MD-SNN for the MNIST dataset. The number of output neurons which are used to count the output spikes is the same as the number of classes.

Twenty iterations are performed on the training set. The initial weights of the synaptic connections are positive numbers less than 10^{-5}. The threshold of the neuron is $V_{\text{rest}} = 0.96$, and the initial value of the learning rate is set as $\lambda = 0.0005$. As the number of iterations increases, the learning rate decreases. The formula is as follows:

$$\lambda_{i+1} = \lambda_1 \sqrt{\frac{1}{n_{\text{i}}}} \tag{9}$$

where n_{i} represents the number of iterations.

Figs. 5 and 6 show the accuracy of training results and test results with vision windows of different sizes and different randomly selected spike ratios of the whole spike train, respectively. To keep the features of patterns, we have tested different sizes of convolution windows ($n = 1, 3, 5, 7, 9$). From the results, it can be seen that $n = 7$ is the most proper size. It is large enough for extracting and keeping the features of MNIST images. When p is too small, the sub-spike trains would not represent the patterns completely. However, the larger p value may make the sub-spike trains contain more features of patterns, which makes learning hard. $p = 0.4$ is the best ratio of spikes selected for MNIST images. When $n = 7$ and $p = 0.4$, the model obtains the best accuracy of 90.49% on the training set, and the accuracy of the test set is 90.44%.

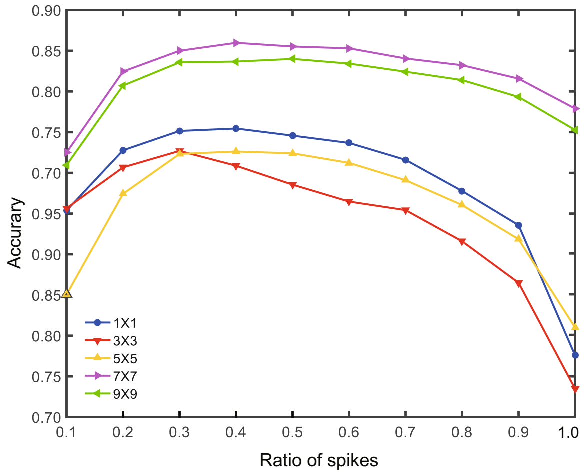

Fig. 5 The accuracy of training results with vision windows of different sizes n and different ratios p of randomly selected spikes of MD-SNN on MNIST

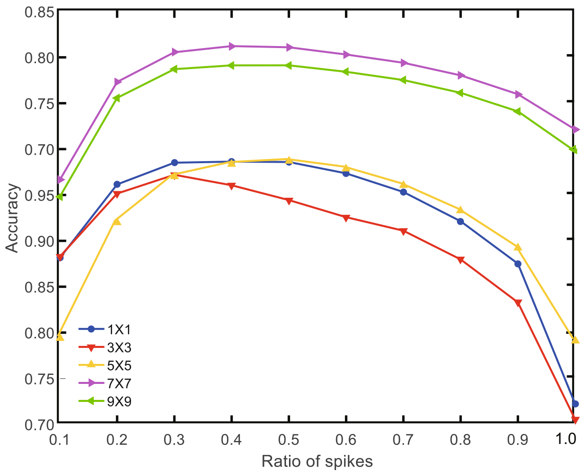

Fig. 6 The accuracy of test results with vision windows of different sizes n and different ratios p of randomly selected spikes of MD-SNN on MNIST

4.1 Comparison with other SNNs

The proposed MD-SNN structure achieves better classification performance on the MNIST dataset than many previous methods, as can be seen from Table 2. In MD-SNN, more features of spike trains are used to do classification, and multiple neurons, the number of which is the same as that of other structures, are used to make a decision. The strategy of the proposed structure can help SNN improve the performance.

Table 2 Comparison between different SNNs on MNIST dataset classification

Structure	Unsupervised/Supervised	Accuracy
Minimal SNN (Tavanaei and Maida, 2015)	Supervised	75.93%
Spiking RBM (Merolla et al., 2011)	Supervised	89.00%
Dendritic neurons (Hussain et al., 2014)	Supervised	90.30%
MD-SNN (our method)	Supervised	90.44%

4.2 Comparison with CNN

In this study, a general CNN structure for classifying the MNIST dataset is selected to compare with our proposed MD-SNN structure (Table 3). The CNN structure consists of four layers: a convolution layer, a pooling layer and two full connection layers.

Table 3 shows that although the classification accuracy of the MD-SNN structure is less than that of the CNN structure on the MNIST dataset currently, the numbers of neurons and layers are smaller, and the calculation method is simpler. According to

Table 3 Comparison of structure and accuracy between MD-SNN and CNN

Network feature	Value	
	MD-SNN	CNN
Number of layers	3	4
Number of neurons	210	7210
Test accuracy	90.44%	98.19%

the comparison between the input spikes and neural membrane potential threshold, the spiking neuron gives the output spike train. It is easier to achieve and consumes less hardware resource.

The basic idea of SNN synaptic weight calculation is based on the difference between the actual output spike train and the expected output spike train, which is used to make adjustments. Since the SNN information is transmitted by the spike trains, the comparison between the spike trains and the adjustment of synaptic weights is quite different from the numerical calculation of the network structure. The traditional back-propagation method cannot be effectively applied. Therefore, how to construct the multi-layer SNN structure is still an open problem. The features extracted by the single-layer network are very limited. If using a single neuron to judge the output, the training accuracy of SNN under the same parameters with a 7 × 7 vision window can reach only 82.59% on the MNIST dataset, and the test accuracy is 81.67%. Although there is only one layer of hidden neurons of the MD-SNN structure, we can improve the feature extraction ability of the network by increasing the number of single-layer neurons. Experiments show that increasing the monolayer neurons can lead to extraction of more image features to improve the final classification accuracy.

4.3 Flip invariance and rotation invariance

In the real world, objects can appear at any position and any angle in an image. Sometimes things may flip up and down, flip left and right, or even rotate in a picture. However, we can still recognize what the flipped or rotated images represent; for example, no matter how an apple is positioned, people always know it is an apple. We call these phenomena flip invariance and rotation invariance.

As a bionic neural network model, the encoding method of MD-SNN based on the receptive field is a centrosymmetry and rotational symmetry coding method. The neurons just depend on the time of spikes produced to analyze the information of the image, and do not rely on the spatial location of pixels in the image. When the images have been flipped up and down or flipped left and right, or even rotated by 90° (Fig. 7), the accuracy of MD-SNN will not change. However, when the flipped or rotated images are tested in a CNN model, the accuracy will sharply decline. We have used these three types of actions of the MNIST test set to test our trained MD-SNN and a trained CNN. The results are shown in Fig. 8.

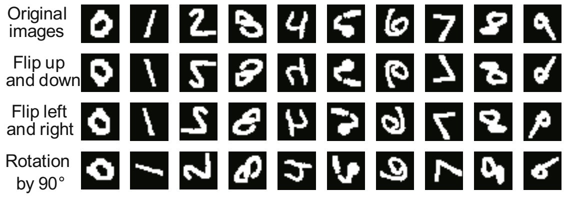

Fig. 7 The original digital images and the digital images flipped up and down, flipped left and right, and rotated by 90°

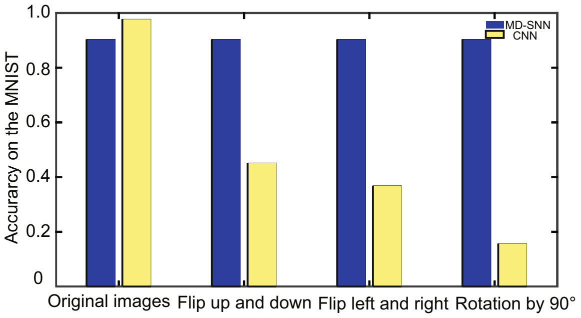

Fig. 8 The comparison of flip invariance and rotation invariance between MD-SNN and CNN

Fig. 8 shows that, no matter whether the test pictures are flipped or rotated, the accuracy of the trained MD-SNN remains unchanged. However, when the pictures of the MNIST test set are flipped up and down, the accuracy of the trained CNN drops to 45.22%, and the CNN can just recognize the symmetrical up and down digits, such as 0, 1, 3, and 8. The accuracy of the trained CNN drops to 36.88% when the pictures are flipped left and right, and the CNN can just recognize the symmetrical left and right digits, such as 0, 1, and 8. The trained CNN can just recognize the digit 0, which is close to central symmetry, if the pictures are rotated by 90°.

The experimental results show that the proposed MD-SNN has a favorable invariance for image flip and rotation. These characteristics come from the symmetry of the receptive field and the time invariance of spike trains.

4.4 Small sample learning

Most artificial neural networks (ANNs) need a large number of training sets for learning. However, in practice, it is hard to find enough data with labels for training, such as medical images, electroencephalogram (EEG) data, and functional magnetic resonance imaging (fMRI) data. Compared with CNN, even with hundreds of training samples, MD-SNN is able to achieve an acceptable accuracy rate on the test set (Table 4). This makes MD-SNN more suitable on small sample learning for classification than CNN.

Table 4 The accuracy comparison between MD-SNN and CNN with different sizes of training set

Training set size	Value	
	MD-SNN	CNN
200	61.42%	9.80%
500	73.84%	9.82%
1000	80.15%	10.28%
2000	85.22%	14.92%
5000	88.02%	68.02%
10 000	89.05%	77.87%
60 000	90.44%	98.19%

5 Discussion

The spiking neural network, which consists of the spiking neuron model, is considered as a more biologically authentic neural network. The spike train is used to represent and process information. It integrates many kinds of information such as time, space, frequency, and phase. Compared with the traditional ANN based on pulse frequency coding information, SNN has more computational potential and can simulate all kinds of neural signals and arbitrarily continuous functions. It is very suitable for biological neural network signal processing (Tang et al., 2012). SNN is an effective tool for time and space information processing (Dora et al., 2015a,b; Xie et al., 2017).

However, so far, the study on SNN in the field of cognitive and pattern recognition has been very limited. Further research should be focused on effectively encoding the external stimulus signal as a spike train and building a multi-layer network structure. When encoding the external input stimulus signal into a spike train, the input information needs to be integrated and combined with the temporal information. The temporal information is the most significant characteristic of the spike train, and it has advantages over numerical operation. However, to the best of our knowledge, so far there has been no effective pulse encoding method.

In multi-layer SNN construction, many problems need to be resolved. The first problem is how to calculate the error. The error in numerical calculation can be obtained by subtracting the actual output from the expected output. However, the actual output may be identical to or different from the expected output. Even if some researchers have tried to calculate the difference between the spike trains by defining the distance between spike trains (Victor and Purpura, 1996), it is still difficult to apply the new distance definition to error transmission between multi-layer networks.

The second problem is how to realize the gradient descent when there are only two methods to remove the error of the output spike train. It is different from the digital values for which we can use the gradient to correct the parameters of the model. The parameters of SNNs are currently modified according to the expected output to adjust the synaptic weights to change the actual pulse output, making the actual output closer to the desired output. The error between the layers cannot be transferred, because the expected output appears only in the final output, and the desired output of hidden layers cannot be calculated. The spike train of the final output can determine only right or wrong; however, such information cannot be effectively transmitted in multi-layer networks. Some researchers have successfully constructed the structure of multi-layer networks on SNNs through adjustment of the network structure (Zenke and Ganguli, 2017). However, the application of these kinds of network structures is very limited and the computational complexity is very high. Because the features that can be extracted from the single-layer network are always finite and simple, it is hard to effectively process complex features from the single-layer network. In summary, how to establish a multi-layer SNN structural is still not effectively addressed. This is a key factor limiting the accuracy of SNNs.

6 Conclusions

In this study, we established a new structure of SNN, named MD-SNN, which includes three stages: encoding, learning, and readout. In the encoding stage, the characteristic of the biological vision system and the respective field of the DoG model are used to encode the images. The main difference between MD-SNN and other SNN structures lies in the learning stage. Here we use a random selection strategy to generate more spike trains, including more features of images, and use multiple groups of spiking neurons to learn these features. Comparing MD-SNN with CNN, we find that MD-SNN uses less resource to classify images. It shows better flip invariance and rotation invariance, and it is more suitable for small sample learning.

SNN has great potential for research and development. In the future, we will adopt various operators to enrich our model and extend MD-SNN to establish a multi-layer spiking neural network.

References

Al-Amri SS, Kalyankar NV, Khamitkar SD, 2010. Image segmentation by using edge detection. *Int J Comput Sci Eng*, 2(3):804-807.

Berry MJII, Meister M, 1998. Refractoriness and neural precision. Proc Conf on Advances in Neural Information Processing Systems 10, p.110-116.

Bi GQ, Poo MM, 2001. Synaptic modification by correlated activity: Hebb's postulate revisited. *Ann Rev Neurosci*, 24(1):139-166. https://doi.org/10.1146/annurev.neuro.24.1.139

Brette R, Gerstner W, 2005. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J Neurophysiol*, 94(5):3637-3642. https://doi.org/10.1152/jn.00686.2005

Burt P, Adelson E, 1983. The laplacian pyramid as a compact image code. *IEEE Trans Commun*, 31(4):532-540. https://doi.org/10.1109/TCOM.1983.1095851

Canny J, 1986. A computational approach to edge detection. *IEEE Trans Patt Anal Mach Intell*, 8(6):679-698. https://doi.org/10.1109/TPAMI.1986.4767851

Coates A, Ng A, Lee H, 2011. An analysis of single-layer networks in unsupervised feature learning. Proc 14th Int Conf on Artificial Intelligence and Statistics, p.215-223.

Dasgupta S, Stevens CF, Navlakha S, 2017. A neural algorithm for a fundamental computing problem. *Science*, 358(6364):793-796. https://doi.org/10.1126/science.aam9868

Dora S, Suresh S, Sundararajan N, 2015a. A sequential learning algorithm for a spiking neural classifier. *Appl Soft Comput*, 36:255-268. https://doi.org/10.1016/j.asoc.2015.06.062

Dora S, Sundaram S, Sundararajan N, 2015b. A two stage learning algorithm for a growing-pruning spiking neural network for pattern classification problems. Int Joint Conf on Neural Networks, p.1-7. https://doi.org/10.1109/ijcnn.2015.7280592

Dora S, Subramanian K, Suresh S, et al., 2016. Development of a self-regulating evolving spiking neural network for classification problem. *Neurocomputing*, 171:1216-1229. https://doi.org/10.1016/j.neucom.2015.07.086

Dora S, Suresh S, Sundararajan N, 2017. Online meta-neuron based learning algorithm for a spiking neural classifier. *Inform Sci*, 414:19-32. https://doi.org/10.1016/j.ins.2017.05.050

Fukushima K, 1980. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern*, 36(4):193-202. https://doi.org/10.1007/bf00344251

Gerstner W, Kistler W, 2002. Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9780511815706

Ghosh Dastidar S, Adeli H, 2007. Improved spiking neural networks for eeg classification and epilepsy and seizure detection. *Integr Comput Aided Eng*, 14(3):187-212.

Gilbert CD, Wiesel TN, 1992. Receptive field dynamics in adult primary visual cortex. *Nature*, 356(6365):150-152. https://doi.org/10.1038/356150a0

Gütig R, Sompolinsky H, 2006. The tempotron: a neuron that learns spike timing-based decisions. *Nat Neurosci*, 9(3):420-428. https://doi.org/10.1038/nn1643

Hannun AY, Case C, Casper J, et al., 2014. Deep speech: Scaling up end-to-end speech recognition. https://arxiv.org/abs/1412.5567

He KM, Zhang XY, Ren SQ, et al., 2016. Deep residual learning for image recognition. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.770-778. https://doi.org/10.1109/CVPR.2016.90

Hodgkin AL, Huxley AF, 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol*, 117(4):500-544. https://doi.org/10.1113/jphysiol.1952.sp004764

Hubel DH, Wiesel TN, 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol*, 160(1):106-154. https://doi.org/10.1113/jphysiol.1962.sp006837

Hussain S, Liu SC, Basu A, 2014. Improved margin multi-class classification using dendritic neurons with morphological learning. IEEE Int Symp on Circuits and Systems, p.2640-2643. https://doi.org/10.1109/iscas.2014.6865715

Izhikevich EM, 2001. Resonate-and-fire neurons. *Neur Networks*, 14(6-7):883-894. https://doi.org/10.1016/s0893-6080(01)00078-8

Izhikevich EM, 2003. Simple model of spiking neurons. *IEEE Trans Neur Networks*, 14(6):1569-1572. https://doi.org/10.1109/tnn.2003.820440

Izhikevich EM, 2004. Which model to use for cortical spiking neurons? *IEEE Trans Neur Networks*, 15(5):1063-1070. https://doi.org/10.1109/tnn.2004.832719

LeCun Y, Bengio Y, Hinton G, 2015. Deep learning. *Nature*, 521(7553):436-444. https://doi.org/10.1038/nature14539

Legenstein R, Naeger C, Maass W, 2006. What can a neuron learn with spike-timing-dependent plasticity? *Neur Comput*, 17(11):2337-2382. https://doi.org/10.1162/0899766054796888

Ma YQ, Wu H, Zhu MJ, et al., 2017. Reconstruction of visual image from functional magnetic resonance imaging using spiking neuron model. *IEEE Trans Cogn Dev Syst*, in press. https://doi.org/10.1109/tcds.2017.2764948

Maass W, 1997. Networks of spiking neurons: the third generation of neural network models. *Neur Networks*, 10(9):1659-1671. https://doi.org/10.1016/s0893-6080(97)00011-7

Masquelier T, Guyonneau R, Thorpe SJ, 2009. Competitive stdp-based spike pattern learning. *Neur Comput*, 21(5):1259-1276. https://doi.org/10.1162/neco.2008.06-08-804

Merolla P, Arthur J, Akopyan F, et al., 2011. A digital neurosynaptic core using embedded crossbar memory with 45pj per spike in 45nm. IEEE Custom Integrated Circuits Conf, p.1-4. https://doi.org/10.1109/cicc.2011.6055294

Ponulak F, Kasiński A, 2010. Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. *Neur Comput*, 22(2):467-510. https://doi.org/10.1162/neco.2009.11-08-901

Rodieck RW, 1965. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vis Res*, 5(11):583-601. https://doi.org/10.1016/0042-6989(65)90033-7

Schmidhuber J, 2015. Deep learning in neural networks: An overview. *Neur networks*, 61:85-117. https://doi.org/10.1016/j.neunet.2014.09.003

Sobel I, 2014. History and definition of the sobel operator. https://www.scribd.com/document/271811982/History-and-Definition-of-Sobel-Operator

Tang H, Yu Q, Tan KC, 2012. Learning real-world stimuli by single-spike coding and tempotron rule. Int Joint Conf on Neural Networks, p.1-6. https://doi.org/10.1109/ijcnn.2012.6252369

Tavanaei A, Maida AS, 2015. A minimal spiking neural network to rapidly train and classify handwritten digits in binary and 10-digit tasks. *Int J Adv Res Artif Intell*, 4(7):1-8. https://doi.org/10.14569/ijarai.2015.040701

Thorpe S, Delorme A, van Rullen R, 2001. Spike-based strategies for rapid processing. *Neur Netw*, 14(67):715-725. https://doi.org/10.1016/s0893-6080(01)00083-1

Victor JD, Purpura KP, 1996. Nature and precision of temporal coding in visual cortex: a metric-space analysis. *J Neurophysiol*, 76(2):1310-1326. https://doi.org/10.1152/jn.1996.76.2.1310

Wade JJ, Mcdaid LJ, Santos JA, et al., 2010. SWAT: a spiking neural network training algorithm for classification problems. *IEEE Trans Neur Networks*, 21(11):1817-1830. https://doi.org/10.1109/TNN.2010.2074212

Xie XR, Qu H, Yi Z, et al., 2017. Efficient training of supervised spiking neural network via accurate synaptic-efficiency adjustment method. *IEEE Trans Neur Networks*, 28(6):1411-1424. https://doi.org/10.1109/tnnls.2016.2541339

Yeomans JS, 1979. The absolute refractory periods of self-stimulation neurons. *Phys Behav*, 22(5):911-919. https://doi.org/10.1016/0031-9384(79)90336-6

Yu Q, Tang HJ, Tan KC, et al., 2013. Rapid feedforward computation by temporal encoding and learning with spiking neurons. *IEEE Trans Neur Networks*, 24(10):1539-1552. https://doi.org/10.1109/TNNLS.2013.2245677

Yu Q, Tang HJ, Tan KC, et al., 2014. A brain-inspired spiking neural network model with temporal encoding and learning. *Neurocomputing*, 138:3-13. https://doi.org/10.1016/j.neucom.2013.06.052

Zenke F, Ganguli S, 2017. Superspike: supervised learning in multi-layer spiking neural networks. https://arxiv.org/abs/1705.11146