Neurocomputing 219 (2017) 333–349

Contents lists available at ScienceDirect

Neurocomputing

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

Supervised learning in spiking neural networks with noise-threshold☆

Malu Zhang[a], Hong Qu[a,*], Xiurui Xie[a], Jürgen Kurths[b,c,d]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, PR China
[b] Potsdam Institute for Climate Impact Research (PIK), 14473 Potsdam, Germany
[c] Department of Physics, Humboldt University, 12489 Berlin, Germany
[d] Institute for Complex System and Mathematical Biology, University of Aberdeen, Aberdeen AB24 3UE, United Kingdom

ARTICLE INFO

Communicated by Dr. A. Belatreche

Keywords:
Spiking neurons
Noise-threshold
Supervised learning
Spiking neural networks (SNNs)
Anti-noise capability

ABSTRACT

With a similar capability of processing spikes as biological neural systems, networks of spiking neurons are expected to achieve a performance similar to that of living brains. Despite the achievement of spiking neuron based applications, most of them assume noise-free condition for learning and testing. This assumption, though fairly general, ignores the fact that noise widely exists in spiking neural networks (SNNs) and the neural response can be significantly disturbed by noise. Therefore, how to deal with noise is an important issue in the applications of SNNs. Here, by analyzing strategies employed to make spiking neurons robust to noise, also inspired by biological neurons, we propose a strategy that train spiking neurons with a dynamic firing threshold named noise-threshold. The noise-threshold can be applied by the existing supervised learning methods to improve the noise tolerance of them. Experimental results show that, with a combination of noise-threshold, the anti-noise capability of the existing supervised learning methods improves significantly, and the trained neuron can precisely and reliably reproduce target sequences of spikes even under highly noisy conditions. More importantly, the SNNs-based computational model equipped with a noise-threshold is more robust and can achieve a good performance even with different types of noise. Therefore, the noise-threshold is significant to practical applications and theoretical researches of SNNs.

1. Introduction

As the emerging evidence of precise spike-timing neural activities has been observed in many brain regions, including the retina [1–3], the lateral geniculate nucleus [4] and the visual cortex [5], the view that information is represented by explicit timing of spikes rather than mean firing rate has received increasing attention [6,7]. These findings have led to a new way of simulating neural networks based on spiking neurons which encode information by the firing times of spikes [8–10]. Theoretical analysis indicates that neural networks of spiking neurons can arbitrarily approximate any continuous function, and furthermore, it has been demonstrated that networks of spiking neurons are computationally more powerful than sigmoidal neurons [11–14].

Spiking neurons can be used to construct a network that stores the information in synapses between neurons [8,15,16]. Changing the synaptic efficiency could result in acquiring new information. The modified way of synaptic efficiency is determined by learning algorithms [17]. Recently, learning schemes focusing on processing spatio-temporal spikes in a supervised manner have been widely studied, and many supervised learning algorithms have been proposed. These methods can broadly be subdivided into two classes: spike-driven methods and membrane potential-driven methods. Synaptic adaption in spike-driven methods is driven by the error between the desired and actual output spikes. The most popular one of spike-driven methods is remote supervised method (ReSuMe) [18], which is composed of two weight update processes: (1) strengthening synaptic weights by STDP based on input spike trains and a desired output spike train; (2) weakening the synaptic weights by anti-STDP based on input spike trains and an actual output spike train. To enhance the learning performance of ReSuMe, delay learning ReSuMe (DL-ReSuMe) [19] and Multiple DL-ReSuMe (Multi-DL-ReSuMe) [20] are proposed to merge the delay shift approach and ReSuMe-based weight adjustment. Simulation results have shown that the proposed DL-ReSuMe and Multi-DL-ReSuMe achieve learning accuracy and learning speed improvements compared with ReSuMe. Membrane potential-driven methods emerged recently with the typical examples of perceptron-based spiking neuron learning rule (PBSNLR) [21] and High-Threshold Projection (HTP) [22]. They perform a perceptron classification on discretely sampled time points of the voltage, with the aim to keep the membrane potential below a firing threshold for all non-spike

☆ This work was supported in part by the National Science Foundation of China under Grants 61273308, 61370073 and 61573081.
* Corresponding author.
E-mail address: hongqu@uestc.edu.cn (H. Qu).

http://dx.doi.org/10.1016/j.neucom.2016.09.044
Received 10 October 2015; Received in revised form 11 May 2016; Accepted 17 September 2016
Available online 24 September 2016
0925-2312/

times and to ensure a threshold crossing at the desired output times [23]. In addition to ReSuMe and PBSNLR, there are still many other efficiency methods with the goal of training spiking neurons to generate desired sequences of spikes [24–29].

Despite remarkable progress has been made in learning algorithms of SNNs, many researches and applications just consider the simple case where the neurons are trained and tested under noise free condition. However, noise sources widely exist in SNNs, and can be distinguished as intrinsic and extrinsic ones [8]. The main sources of intrinsic noise are: (1) Johnson noise due to thermal fluctuations of membrane resistance [30]; (2) open-close fluctuations of the ion channels, which may originate from ion concentration fluctuations [31]. Apart from intrinsic noise sources, extrinsic sources are mainly due to signal transmission and network effects. It has been shown that networks of excitatory and inhibitory neurons with fixed random connectivity can produce highly irregular spike trains even in the absence of any sources of noise [32,33]. The above noise sources will certainly affect the timing accuracy and reliability of SNNs [34–36], especially when we apply SNNs to hardware models [37,38].

Though there are also various noise sources in biological neural networks, the brain responds rapidly and reliably to fine details in sensory input. Experimental results provide evidence that the nervous system can deal with noise to produce accurate and reliable response [39–41]. However, the exact mechanisms employed remain unknown. To increase some anti-noise capability of the trained SNNs, most of the supervised learning methods adopt the strategy that train spiking neurons in the presence of noise (i.e., noisy training). However, the neuron trained under noisy conditions demonstrates relative high robustness to noise only in response to the stimuli used during the training, and the neuron responds highly unreliably to other stimuli [18]. Recently, a new plasticity rule called Membrane Potential Dependent Plasticity (MPDP) has been proposed [23]. The sensitivity of the MPDP to the subthreshold membrane potential during the training allows a robust memory retrieval after learning even in the presence of noise. However, the learning efficiency of MPDP is relatively low, and its anti-noise capability still could be improved.

In this paper, we propose a dynamic firing threshold named noise-threshold for training spiking neurons. The noise-threshold could be applied to the existing supervised learning methods to improve the anti-noise capability. ReSuMe and PBSNLR, as the most typical representative learning methods in SNNs, are selected to illustrate how a noise-threshold could be combined with supervised learning rules of SNNs. In a set of experiments, we demonstrate that: (1) supervised learning methods combined with noise-threshold maintain the advantages of the original algorithms that enable us to train spiking neurons with a high accuracy and efficiency; (2) the neuron trained with noise-threshold can precisely and more reliably reproduce target firing patterns even under a relatively high level of noise; (3) a SNNs-based computational model equipped with noise-threshold is more robust, which is effective to deal with noisy data and can make a proper decision even under highly noisy conditions. Therefore, noise-threshold has wide application prospects and can be widely used in SNNs-based models to improve their robustness.

The rest of this paper is organized as follows. In Section 2, the learning rules of PBSNLR and ReSuMe are introduced. The basic idea of noise-threshold and how noise-threshold could be combined with supervised learning methods of SNNs are presented in Section 3. In Section 4, some experiments are given to investigate the anti-noise capability of noise-threshold. In addition, we explore the effects of parameters involved in noise-threshold. Discussion of our method is presented in Section 5. Conclusion and future work are presented in Section 6.

2. Neuron model and supervised learning rules of spiking neurons

ReSuMe and PBSNLR are the most typical examples of supervised learning methods in SNNs. We take them to illustrate how noise-threshold could be combined with existing supervised learning methods of SNNs. Therefore, in this section, the neuron model used in our research and the basic ideas of ReSuMe and PBSNLR will be introduced.

2.1. Neuron model

The leaky integrate-and-fire (LIF) model is considered in this paper. The dynamics of each neuron evolves according to the following equation

$$V(t) = \sum_j \omega_{ij} \sum_f K(t - t_j^f) + V_{rest} \tag{1}$$

where t_j^f is the fth spike of presynaptic neuron j, and ω_{ij} is the synaptic

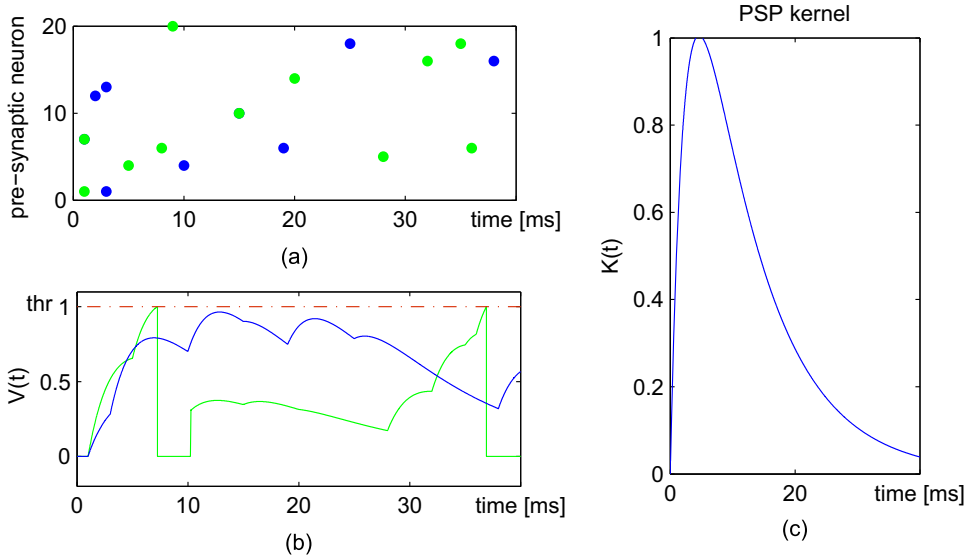


Fig. 1. Dynamic of LIF neuron response. (a) Examples of input patterns. There are two patterns (blue and green) and each spike which is fired by pre-synaptic neuron is denoted by a dot. (b) Membrane potential traces of the two input patterns. Each colored line corresponds to the pattern with the same color on left (top). Pattern one (in green) fires two spikes and Pattern two (in blue) does not fire. (c) Normalized Postsynaptic kernel K(t) with membrane time constant τ_m = 10 ms and synaptic time τ_s = 2.5 ms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

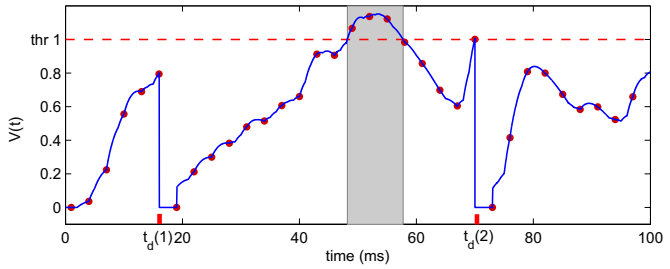


Fig. 2. Illustration of the PBSNLR learning rule. During the learning period, there are two desired output times $t_d(1)$ and $t_d(2)$. The red dots on the membrane potential trace represent the positive and negative samples. All misclassified samples are trained using PBSNLR which depends on two cases: (1) At the positive samples (desired output time), if the membrane potential is below threshold, the synaptic weights should be potentiated; (2) At the negative samples (not desired output time), if membrane potential is above threshold (as shown in shaded area), the synaptic weights should be depressed.

weight from neuron j to neuron i. V_{rest} is the rest potential of the LIF neuron, which is typically set as 0. K denotes the normalized PSP kernel defined as

$$K(t-t_j^f) = V_0 \times \left(\exp\left(\frac{-(t-t_j^f)}{\tau_m}\right) - \exp\left(\frac{-(t-t_j^f)}{\tau_s}\right)\right) \tag{2}$$

where τ_m and τ_s denote the two decay time constants of membrane integration and synaptic current, respectively. V_0 is a parameter, which is used to make the maximum value of unitary PSP to 1. $K(t-t_j^f)$ is a causal filter that only considers spikes $t_j^f \le t$.

The action potential (spike) is triggered if the membrane potential $V(t)$ of the neuron reach the threshold (thr) at time t_r^f:

$$V(t^{f_r}) \ge \text{thr} \quad and \quad \frac{dV(t^{f_r})}{dt} > 0. \tag{3}$$

After firing, the membrane voltage $V(t)$ resets to the rest potential ($V(t) = 0$), and stays at the resting level for a time period R_a, called the refractory period.

The dynamics of the neuron model are illustrated in the left bottom of Fig. 1.

2.2. ReSuMe learning rule

ReSuMe is a spike-driven supervised learning method of SNNs, which aims to minimize the error between the desired and the actual output spikes. According to ReSuMe, the synaptic weights are modified according to the following equation:

$$\frac{d}{dt}w_{io}(t) = [S_d(t) - S_o(t)][a_d + \int_0^\infty W(s)S_i(t-s)ds], \tag{4}$$

where w_{io} is the synaptic weight between the presynaptic neuron i and the postsynaptic synaptic neuron o. s denotes a delay between the presynaptic and postsynaptic firings. $S(t)$ represents the spike train, and $S_d(t)$, $S_o(t)$ and $S_i(t)$ are the target, post-, and presynaptic spike trains, respectively. The spike trains have the following form

$$S(t) = \sum_f \delta(t - t^f), \tag{5}$$

where $f = 1, 2, \ldots$ is the label of the spikes and $\delta(n)$ is a Dirac function with $\delta(n) = 1$ for $n = 0$ (or 0 otherwise). The kernel $W(s)$ defines the shape of a learning window

$$W(s) = A \cdot \exp\left(\frac{-s}{\tau}\right) \tag{6}$$

where A is the maximal magnitude of the learning window and τ denotes the time constant of the learning process. The sign of the error signal $(S_d(t) - S_o(t))$ decides the direction of the synaptic modification. The kernel $a_d + \int_0^\infty W(s)S_i(t-s)ds$ decides the amount of weight change.

2.3. PBSNLR learning rule

The PBSNLR [21] transforms the supervised learning into a classification problem using the perceptron learning rule, with the aim to keep membrane potential below threshold at undesired spike times and to make sure a threshold crossing at desired spike times. The learning rule of PBSNLR can be expressed as

$$w_i^{new} = \begin{cases} w_i^{old} + \beta P_i^t, & \text{if} \quad d_t = 1 \quad \text{and} \quad V(t) < thr \\ w_i^{old} - \beta P_i^t, & \text{if} \quad d_t = 0 \quad \text{and} \quad V(t) \ge thr \\ w_i^{old}, & \text{otherwise} \end{cases} \tag{7}$$

Where w_i is the weight of the ith synapse, and P_i^t is the sum of PSPs induced by all the spikes that have arrived through the ith synapse at t. d_t is a signal to represent whether t is the desired output time. d_t=1 (or d_t=0) means t is the desired (or undesired) output time. $V(t)$ is the membrane potential of the neuron at t. Fig. 2 gives a simple example to illustrate the learning rule of PBSNLR.

3. Noise-threshold

In this section, first we propose the concept of noise-threshold. Then, ReSuMe and PBSNLR are selected as examples to illustrate how noise-threshold could be combined with supervised learning methods

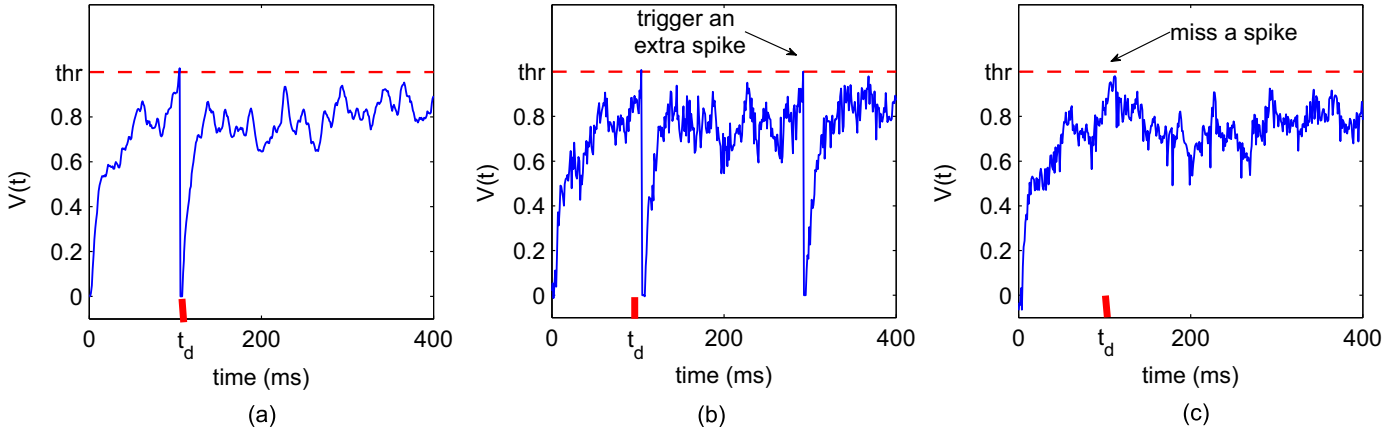


Fig. 3. Precision and reliability of neural response can be disturbed by noise. (a) After training, the trained neuron can emit a spike precisely at t_d in the absence of noise. (b) The fluctuating noise would trigger an extra spike if the membrane potential is close to the threshold at Nt_d. (c) The noise makes the membrane potential below threshold at t_d, then the trained neurons will miss a desired spike.

of SNNs.

3.1. Noise-threshold

The goal of supervised learning in SNNs is that the synaptic weights are adjusted to make the membrane potential reach firing threshold at the desired output time (t_d), otherwise let it below firing threshold at the undesired output time (Nt_d). Original learning methods train spiking neurons with a fixed threshold, resulting in that the response of the trained neuron can be easily disturbed by noise. Fig. 3 depicts two basic ways of noise disturbing the trained neurons..

From Fig. 3, we can find that: (1) if the membrane potential is close to threshold at Nt_d, the probability of triggering a wrong spike will increase. Therefore, to avoid an extra spike, the membrane potential at Nt_d should keep far away from the threshold; (2) to make sure that the neuron will fire nearby t_d, the membrane potential around t_d should be strong enough.

Based on the above analysis, we propose a dynamic firing threshold named noise-threshold for spiking neurons during the learning process. Noise-threshold divides the running time of a spiking neuron into two classes denoted by $\widetilde{t}_d$ and $\widetilde{Nt}_d$:

$$\widetilde{t}_d = \{t | t \in [t_d(i) - \delta,\ t_d(i) + \delta]\} \tag{8}$$

$$\widetilde{Nt}_d = \{t | t \notin [t_d(i) - \delta,\ t_d(i) + \delta]\} \tag{9}$$

where $t_d(i)$ denotes the moment of the ith spike in the desired spike train and δ is a parameter determining the length of $\widetilde{t}_d$ and $\widetilde{Nt}_d$. Fig. 4 gives an example to illustrate $\widetilde{t}_d$ and $\widetilde{Nt}_d$. In the following, the definition of noise-threshold based on $\widetilde{Nt}_d$ and $\widetilde{t}_d$ will be introduced, respectively.

3.1.1. Noise-threshold at $\widetilde{Nt}_d$

After a successful learning with supervised learning methods, the membrane potential will below firing threshold at Nt_d. Therefore, to make the membrane potential far away from threshold, we can set a small value for noise-threshold during learning process as

$$\mathrm{n-thr} = \mathrm{thr} - \eta_1 \quad t \in \widetilde{Nt}_d \tag{10}$$

where $\eta_1 > 0$, and thr is the traditional fixed threshold. To better illustrate how noise-threshold can keep the membrane potential far away from threshold and the role of η_1, we give a simple example in Fig. 5.

What need attention is that noise-threshold is just used for learning process, we test the performance of the trained neuron with traditional fixed threshold. As shown in Fig. 5(c), after training with noise-threshold, the membrane potential keeps far away from threshold to avoid spurious firing.

3.1.2. Noise-threshold at $\widetilde{t}_d$

After a successful learning with supervised learning methods, the membrane potential will reach threshold at t_d. To make the membrane potential strong enough at $\widetilde{t}_d$, we can set a large value for noise-threshold as

$$\mathrm{n-thr} = -a[t - t_d(i)]^2 + \mathrm{thr} + \eta_2, \quad t \in \widetilde{t}_d \tag{11}$$

with constants $\eta_2 > 0$, $a > 0$. To better illustrate how noise-threshold can keep the membrane potential strong enough at $\widetilde{t}_d$ and the roles of η_2 and a, we give a simple example in Fig. 6.

In summary, the noise-threshold function we proposed can be expressed as

$$\mathrm{n-thr} = \begin{cases} \mathrm{thr} - \eta_1, & \text{if} \quad t \in \widetilde{Nt}_d \\ -a[t - t_d(i)]^2 + \mathrm{thr} + \eta_2, & \text{if} \quad t \in \widetilde{t}_d \end{cases} \tag{12}$$

Fig. 7 illustrates the difference between the noise-threshold and traditional fixed threshold. What needs special attention is that we utilize the noise-threshold only for learning processes, and traditional fixed threshold to test the performance of the trained neurons.

3.2. Noise-threshold combined with ReSuMe (N-ReSuMe)

ReSuMe adjusts synaptic weights by desired and actual output spikes. However, noise-threshold is based on the membrane potential level. In order to make a combination of ReSuMe and noise-threshold (N-ReSuMe), we change the spike generating mechanism of the spiking neurons. During the learning process, firing occurs whenever the membrane potential $V(t)$ reaches the noise-threshold instead of threshold. Fig. 8 illustrates the difference of spike generating mechanisms between ReSuMe and N-ReSuMe..

N-ReSuMe merges the actual output spikes generated with noise-threshold, the desired output spikes and ReSuMe-based weight adjustment to update the synaptic weights. According to the N-ReSuMe learning rule, the actual output spikes generated with noise-threshold will cause a depression of synaptic weights to make the membrane potential below the noise-threshold at $\widetilde{Nt}_d$, which leads the membrane potential far away from traditional fixed threshold. In addition, at t_d, the desired output spikes will potentiate the synaptic weights to make the membrane potential equal to the noise-threshold.

3.3. Noise-threshold combined with PBSNLR (N-PBSNLR)

Both PBSNLR and noise-threshold are based on the membrane potential level, to create a combination of noise-threshold and PBSNLR (N-PBSNLR), we change the rule of PBSNLR as: (1) positive samples are misclassified if the membrane potential of the neuron cannot reach the noise-threshold at t_d; (2) negative samples are misclassified if the membrane potential of the neuron exceeds the noise-threshold at Nt_d. Then, all of these misclassified samples are trained using N-PBSNLR. Fig. 9 gives a simple example to illustrate the learning process of N-PBSNLR..

As the noise-threshold is below threshold at $\widetilde{Nt}_d$, the misclassified negative samples in Fig. 9 significantly outnumber those in Fig. 2. These misclassified negative samples will be trained by N-PBSNLR until the membrane potential is below the noise-threshold at $\widetilde{Nt}_d$. On the other hand, the number of misclassified positive samples in Fig. 9 is still larger than in Fig. 2. For example, in Fig. 2, $t_d(2)$ is a correct classification that the membrane potential is above the threshold at the desired output time. However, in Fig. 9, $t_d(2)$ is a misclassified sample. According to N-PBSNLR, synaptic weights need to be modified until the membrane potential reaches the noise-threshold at $t_d(2)$.

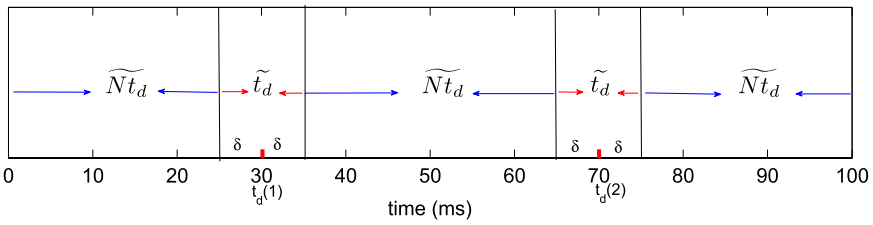


Fig. 4. Illustration of the $\widetilde{t}_d$ and $\widetilde{Nt}_d$. There are two desired output spikes ($t_d(1) = 30$ ms and $t_d(2) = 70$ ms) in the total time duration. The parameter δ which determines the distribution of $\widetilde{t}_d$ and $\widetilde{Nt}_d$ is set as 5 ms.

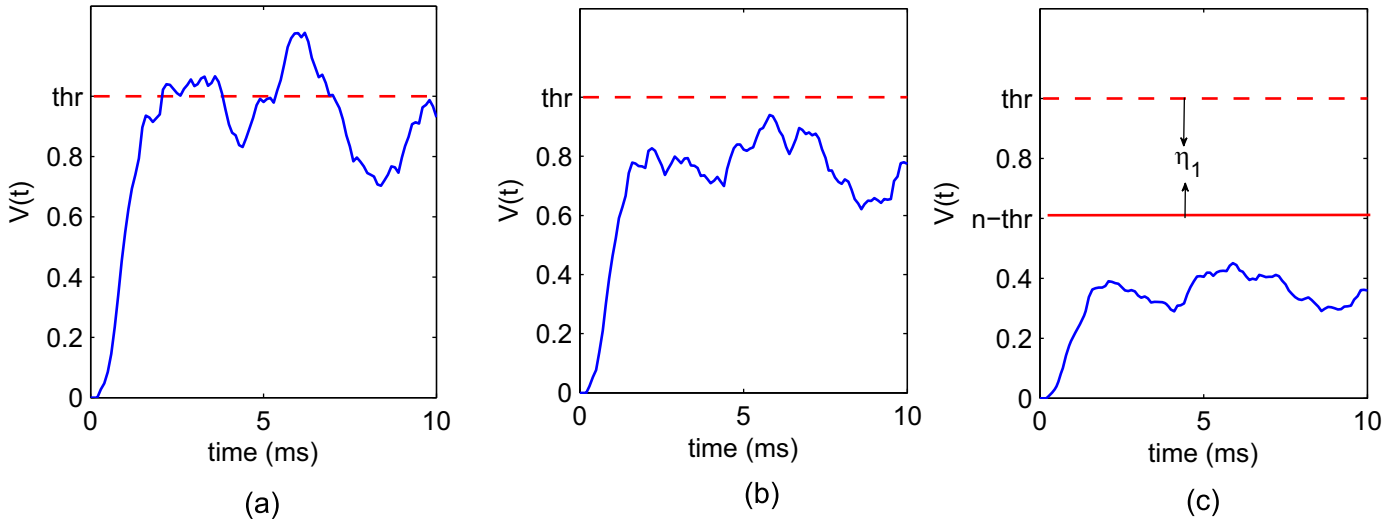


Fig. 5. Relationship between membrane potential and noise-threshold at $\widetilde{Nt}_d$. The dashed line represents the traditional fixed threshold (thr) and the solid line represents the noise-threshold (n-thr). (a) The membrane potential trajectory before learning. The membrane potential is above threshold at $\widetilde{Nt}_d$ (0–10 ms). (b) The membrane potential trajectory after learning with threshold. After a successful learning with threshold, the membrane potential is below threshold. (c) The membrane potential trajectory after learning with noise-threshold. After a successful learning with noise-threshold, the membrane potential is below noise-threshold, which is far away from the threshold with a distance at least η_1. The parameters in this figures are thr=1 mV and η_1=0.4 mV.

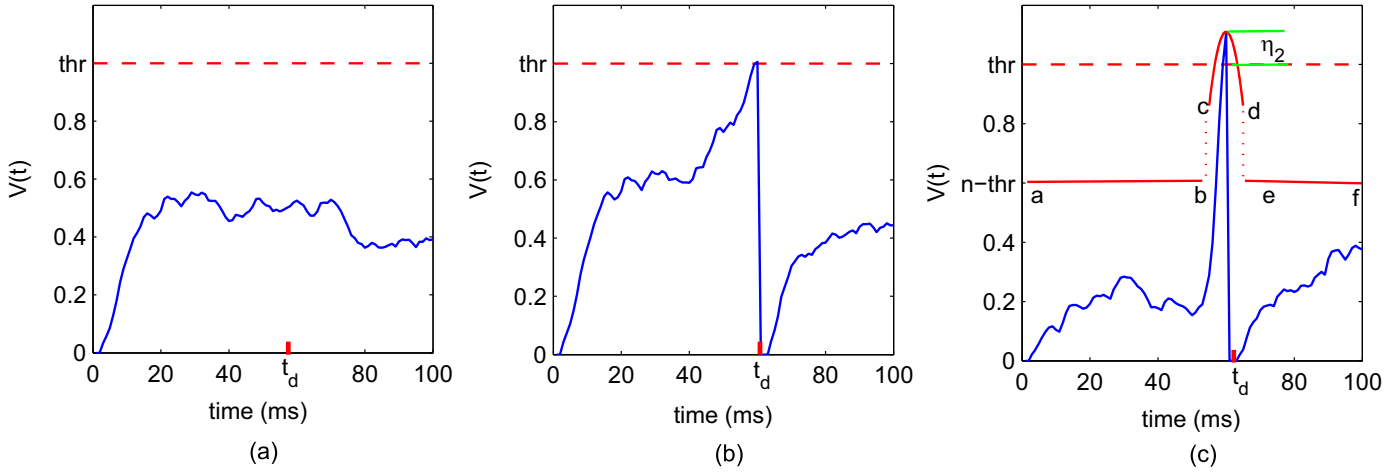


Fig. 6. Relationship between membrane potential and noise-threshold at $\widetilde{t}_d$. The dashed line represents the traditional fixed threshold and the solid line represents the noise-threshold. (a) The membrane potential trajectory before learning. The membrane potential is below threshold at t_d (60 ms). (b) The membrane potential trajectory after learning with threshold (thr). After a successful learning with threshold, the membrane potential will reach threshold at t_d. (c) The membrane potential trajectory after learning with noise-threshold (n-thr). In the period of $\widetilde{t}_d$ (c → d), noise-threshold is defined as Eq. (11), where a decides the opening direction and size of the parabola, and η_2 decides the top of the parabola. After a successful learning, the membrane potential will reach thr+η_2 at t_d. The parameters in these figures are thr=1 mV, δ=5 ms, η_1=0.4 mV, η_2=0.1 mV, and a=0.01.

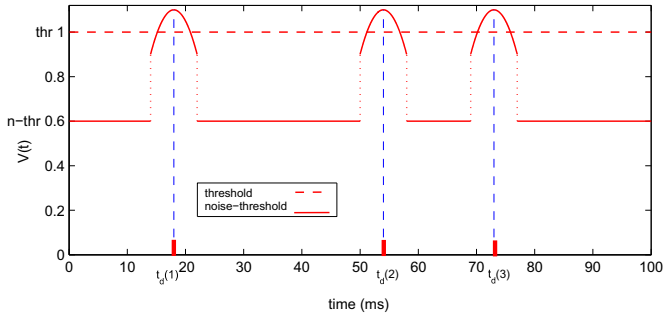


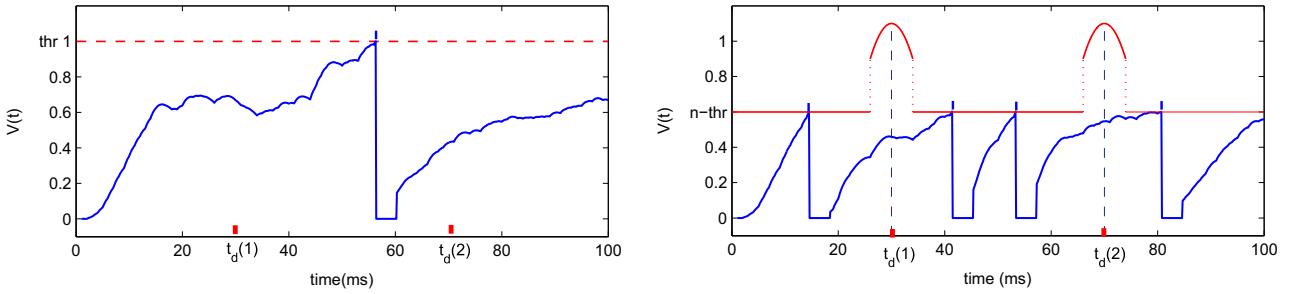
Fig. 7. Comparison between traditional fixed threshold and noise-threshold. $t_d(1)$, $t_d(2)$ and $t_d(3)$ are three desired output times, which are generated according a homogeneous Poisson process with a frequency of 5 Hz. The dashed line represents the traditional fixed threshold and the solid line represents the noise-threshold. The value of noise-threshold is below threshold at $\widetilde{Nt}_d$ and is above threshold at the middle of $\widetilde{t}_d$. The parameters in this figure are thr=1 mV, δ=5 ms, η_1=0.4 mV, η_2=0.1 mV, and a=0.01.

4. Simulations

In this section, several experiments are presented to demonstrate the learning capability of N-ReSuMe and N-PBSNLR. Next, we show that spiking neurons trained with noise-threshold have a stronger anti-noise capability. Then, we investigate the effects of parameters involved in the noise-threshold approach. Finally, several simulations are performed to test the performance of noise-threshold in practical applications.

4.1. Learning sequence of spikes

In this section, we present a set of experiments to demonstrate that spiking neurons trained according to N-ReSuMe and N-PBSNLR are capable of learning and precisely reproducing desired sequences of spikes. A spiking neuron with 400 synaptic inputs is trained to emit a desired sequence of spikes. The length of input and desired output spike trains is 400 ms. Every input and desired output spike train are generated randomly according to a homogeneous Poisson process with



(a) Running process of the neuron trained with ReSuMe. (b) Running process of the trained neuron with N-ReSuMe.

Fig. 8. The difference of spike generating mechanisms between ReSuMe and N-ReSuMe during the learning process. (a) The spike generating mechanism of ReSuMe. The neuron emits a spike when the membrane potential reaches threshold (thr). (b) The spike generating mechanism of N-ReSuMe. The neuron emits a spike when the membrane potential reaches noise-threshold instead of threshold. The parameters in this figure are thr=1 mV, δ=5 ms, η_1=0.4 mV, η_2=0.1 mV, and a=0.01.

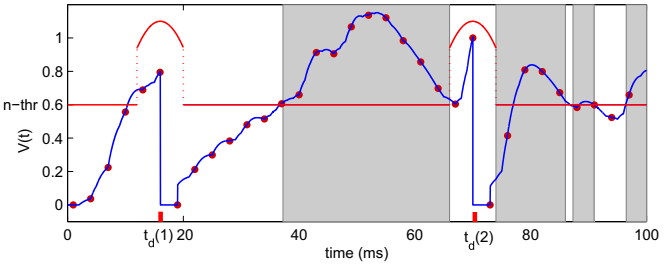


Fig. 9. Illustration of N-PBSNLR learning rule. All misclassified samples are trained using N-PBSNLR which depends on two cases: 1) At positive samples, if the membrane potential is below the noise-threshold, the synaptic weights should be potentiated; (2) At negative samples, if the membrane potential is above the noise-threshold (as shown in shaded area), the synaptic weights should be depressed.

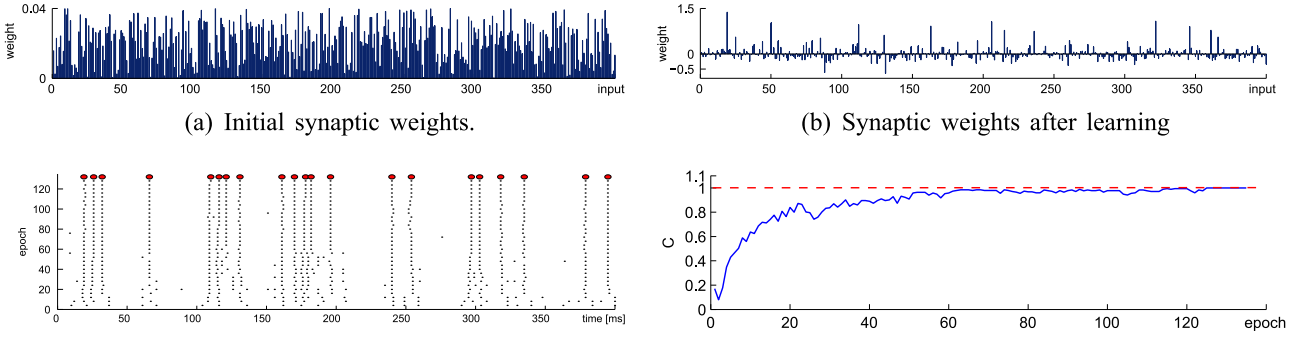
the rate r = 10 Hz and 60 Hz, respectively. The initial synaptic weights are selected as a uniform distribution in the interval [0, 0.04]. The learning process and results of N-ReSuMe and N-PBSNLR are shown in Figs. 10 and 11, respectively.

Fig. 10(a) and (b) represent the synaptic weights before and after learning. Fig. 10(c) illustrates the evolution of the firing patterns generated by the trained neuron in consecutive learning epochs. At first, the actual output spikes are very different from the desired output spikes. After several learning epochs, the gap gets smaller and smaller. At about 125 epochs the actual output spike train becomes the same as the desired one. In order to quantitatively evaluate the learning performance, we use a correlation-based measure C [43] (C is assumed 0 for uncorrelated spike trains and 1 for perfectly matched firing patterns). The measure C plotted as a function of learning epoch is shown in Fig. 10(d), which indicates that the initial value of C is close to 0.2, and C increases to 1 after about 125 learning epochs..

Fig. 11 shows the learning performance of N-PBSNLR. At the beginning, the trained neuron is observed to fire at arbitrary time, resulting in a small C value. During the learning process, the neuron gradually learns to produce spikes at the desired time, and that is also reflected by the increasing of C. The learning accuracy is high after 45 learning epochs, and after about 15 more epochs leads to learning success..

We next investigate the effect of different factors on learning performance of N-ReSuMe and N-PBSNLR. The factors include the spike trains total time duration (Tt), the firing rate of input spike trains (Fin) and the firing rate of output spike trains (Fo). In the following, a neuron with 400 synaptic inputs is considered. The input and desired output spike train are generated randomly according to a homogeneous Poisson process with a frequency Fin and Fo, respectively. For each random pair of input and desired output spike trains, the correlation C is calculated during each learning epoch. Each experiment is repeated for 20 trials for different input and desired output pairs and the median of C, denoted by C_m, is reported. To make a clear comparison, we employ a strategy proposed in [19]. Each value of C_m is replaced by its previous value if there is a drop in the value compared with its previous one to smooth the C_m curve for both N-ReSuMe and N-PBSNLR. The experimental results of N-ReSuMe and N-PBSNLR are shown in Figs. 12 and 13, respectively.

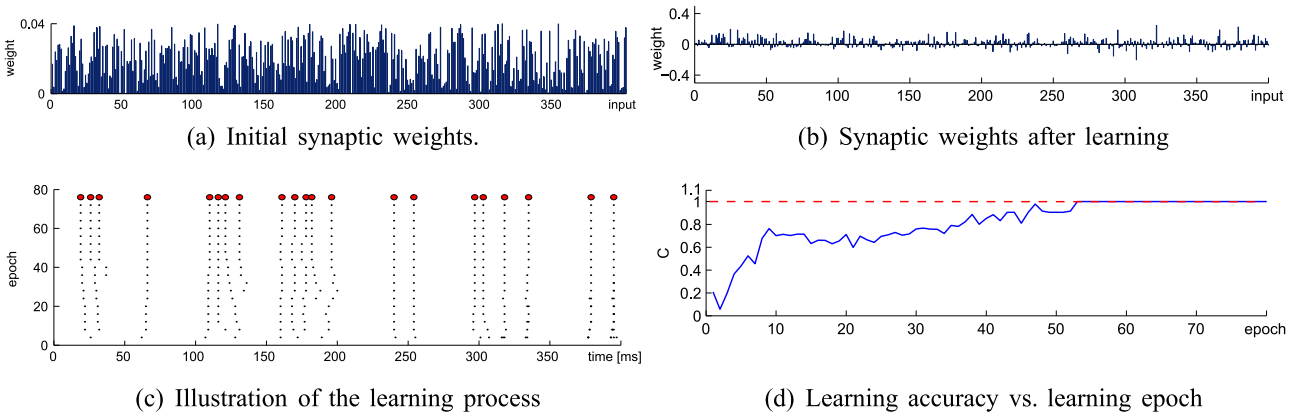
Figs. 12 and 13 show the learning performance of N-ReSuMe and N-PBSNLR, respectively. The experimental results show that the learning accuracy and efficiency of N-ReSuMe and N-PBSNLR are almost same with ReSuMe and PBSNLR, which means that N-ReSuMe and N-PBSNLR maintain the advantages of the original ones that enable us to train spiking neurons with a relatively high accuracy and efficiency.



(a) Initial synaptic weights. (b) Synaptic weights after learning

(c) Illustration of the learning process (d) Learning accuracy vs. learning epoch

Fig. 10. The learning process and performance of N-ReSuMe (a) Initial synaptic weights. (b) Synaptic weights after learning. (c) Illustration of the learning process, which includes the desired output spike train denoted by red ○ and the actual output spike trains after each learning epoch denoted by •. (d) Learning accuracy versus learning epoch. (a) Initial synaptic weights. (b) Synaptic weights after learning. (c) Illustration of the learning process. (d) Learning accuracy vs. learning epoch. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Initial synaptic weights. (b) Synaptic weights after learning

(c) Illustration of the learning process (d) Learning accuracy vs. learning epoch

Fig. 11. The learning process and performance of N-PBSNLR. (a) Initial synaptic weights. (b) Synaptic weights after learning. (c) Illustration of the learning process, which includes the desired output spike train denoted by red ○ and the actual output spike trains after each learning epoch denoted by •. (d) Learning accuracy versus learning epoch. (a) Initial synaptic weights. (b) Synaptic weights after learning. (c) Illustration of the learning process. (d) Learning accuracy vs. learning epoch. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Anti-noise capability

In this part, we will investigate the anti-noise capability of the neuron trained with N-ReSuMe and N-PBSNLR. A neuron with 400 synaptic inputs is considered. The input and desired output spike train are generated randomly according to a homogeneous Poisson process with a frequency *Fin* and *Fo*, respectively. Four training scenarios are considered, i.e., ReSuMe, N-ReSuMe, PBSNLR and N-PBSNLR. After training, the reliability of the target recall is tested against two noise cases: (1) background noise on the membrane potential (2) input jittering noise.

4.2.1. Training and recall with noise on the membrane potential

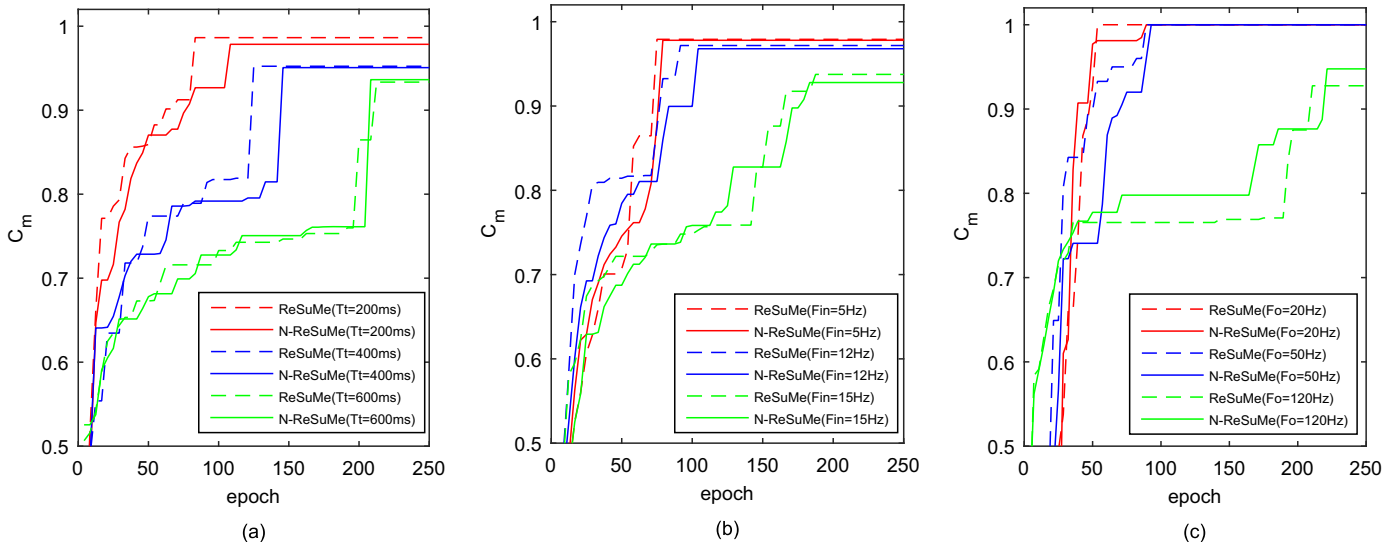
In this case, background voltage noise is considered as the noise source. The trained neuron is subjected to background noise simulated by injecting Gaussian white-noise voltage to the neuron. The mean value of the noise voltage is 0, and its variance σ_b is systematically increased in the range of [0.03, 0.33] mV. Each σ_b is repeated for 20 trials for different input and desired output pairs. These different input and desired output spike trains are generated independently according to a homogeneous Poisson process with a frequency *Fin* and *Fo*, respectively. After training, the reliability of the target recall is tested against background noise. For each value σ_b, a measure C_m of a distance between the desired and actual output trains is calculated. The experimental results are shown in Figs. 14 and 15.

Fig. 14 shows the anti-noise capability of ReSuMe and N-ReSuMe. We observe that the correlation C_m of ReSuMe drops dramatically with the increasing of noise. However, the correlation C_m of the neuron trained by N-ReSuMe almost does not decline and always maintains high values with the increase of σ_b. N-ReSuMe takes values between 0.83 and 1 for all σ_b in the range [0.03, 0.33] mV. These results confirm that the neuron trained with N-ReSuMe is significantly less sensitive to noise.

Fig. 15 reveals that the correlation C_m of both N-PBSNLR and PBSNLR decreases when the C_m gradually increases. The accuracy curve of PBSNLR declines relatively early and quickly. The correlation C_m of N-PBSNLR is higher than that of PBSNLR with a range of noise intensity [0.03, 0.33] mV. That is, the synaptic weights obtained by the N-PBSNLR are more robust to the disruption of noise.

4.2.2. Training and recall with input spike time jitter

In this case, input jittering noise is considered as the noise source. After learning, we jitter the input spike times. The jitter intervals are randomly drawn from a Gaussian distribution with mean 0 and variance $\sigma_j \in [0.3, 3.3]$ ms. In addition, some spikes are randomly canceled (with a probability of 0.1) or added (at the times generated



(a) (b) (c)

Fig. 12. The learning results and the comparison of learning performance between ReSuMe and N-ReSuMe. (a) Effect of the spike trains total time duration (*Tt*). The other parameters *Fin* = 10 Hz, *Fo* = 100 Hz. (b) Effect of the firing rate of input spike trains (*Fin*). The other parameters *Tt*=400 ms, *Fo* = 100 Hz. (c) Effect of the firing rate of output spike trains (*Fo*). The other parameters *Tt*=400 ms, *Fin* = 10 Hz.

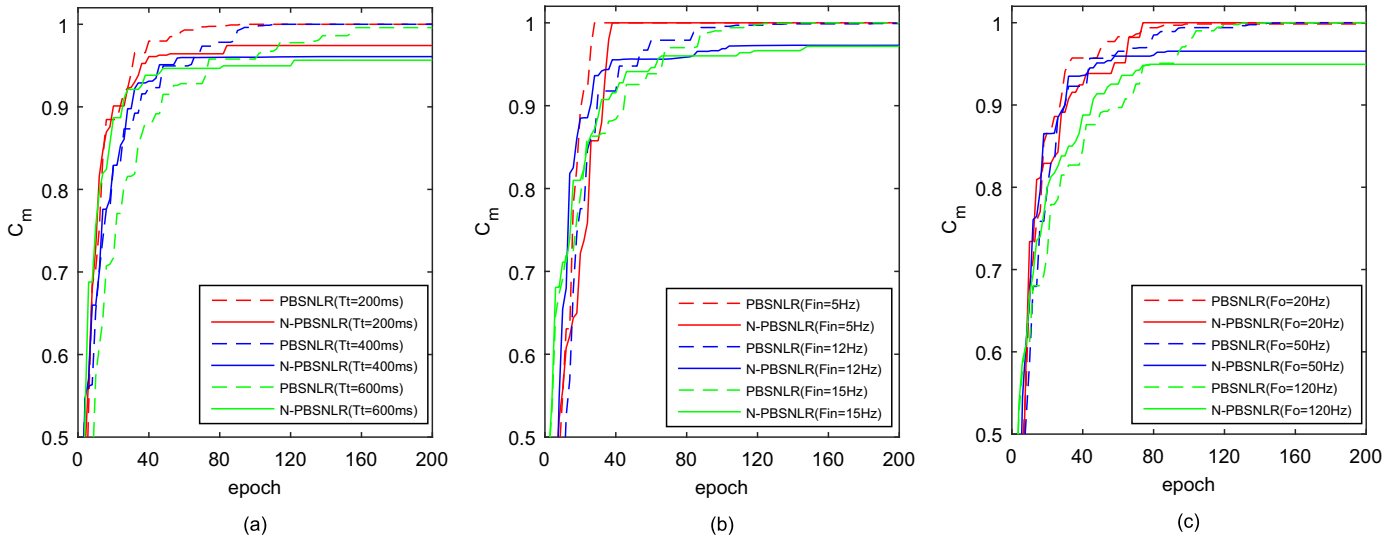


Fig. 13. The learning results and the comparison of learning performance between PBSNLR and N-PBSNLR. (a) Effect of the spike trains total time duration (*Tt*). The other parameters *Fin* = 10 Hz, *Fo* = 100 Hz. (b) Effect of the firing rate of input spike trains (*Fin*). The other parameters *Tt*=400 ms, *Fo* = 100 Hz. (c) Effect of the firing rate of output spike trains (*Fo*). The other parameters *Tt*=400 ms, *Fin* = 10 Hz.

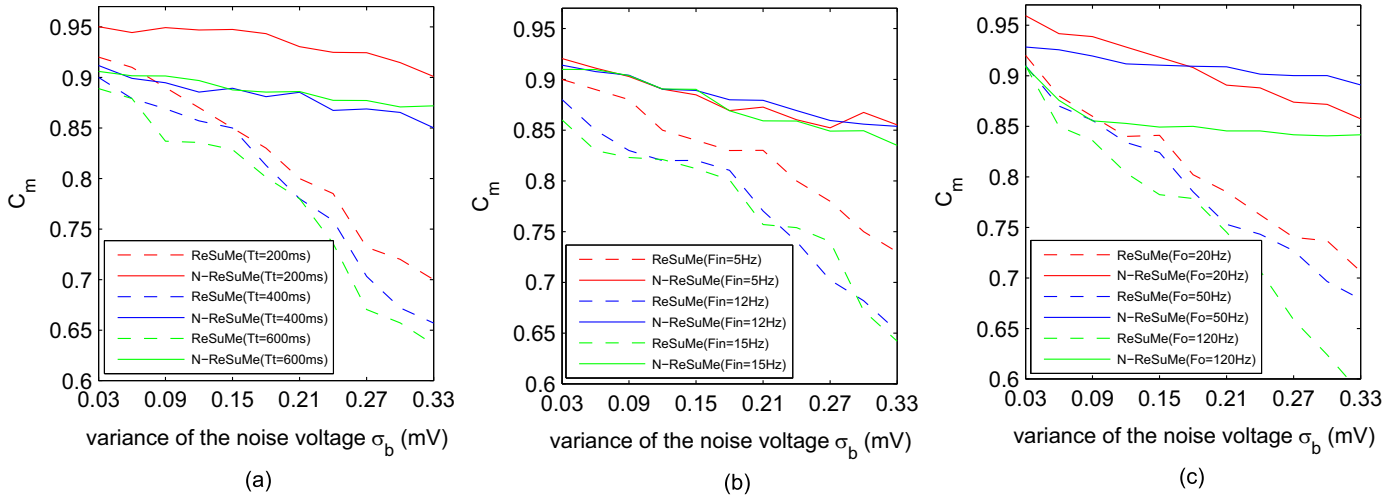


Fig. 14. Anti-noise capability of ReSuMe and N-ReSuMe against background voltage noise. (a) Effect of the spike trains total time duration (*Tt*). The other parameters *Fin* = 10 Hz, *Fo* = 100 Hz. (b) Effect of the firing rate of input spike trains (*Fin*). The other parameters *Tt*=400 ms, *Fo* = 100 Hz. (c) Effect of the firing rate of output spike trains (*Fo*). The other parameters *Tt*=400 ms, *Fin* = 10 Hz.

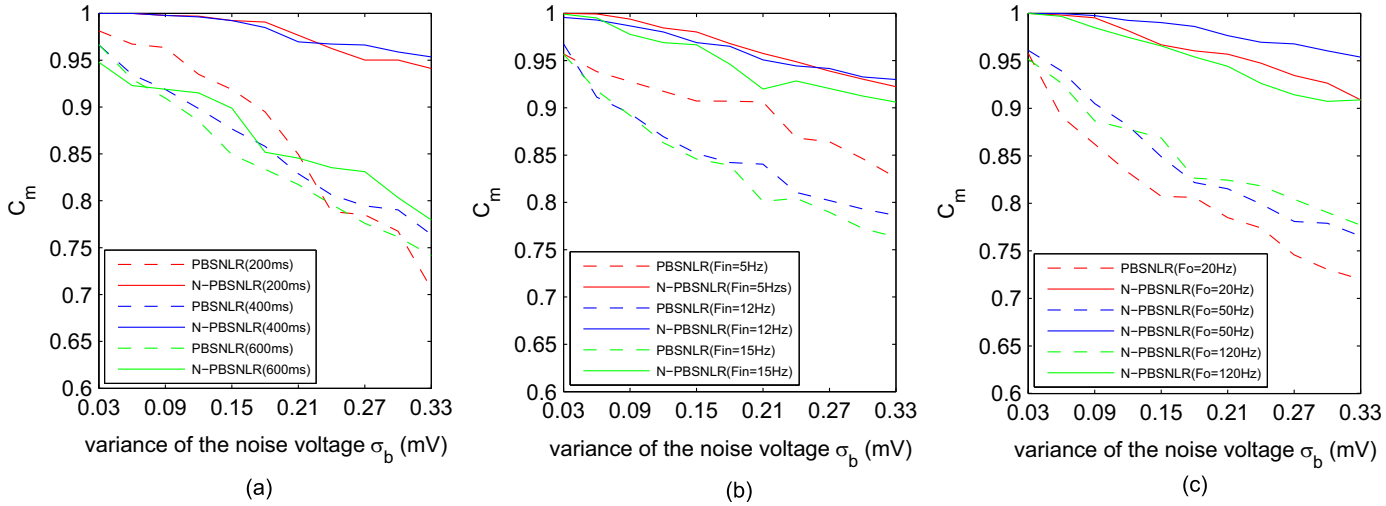


Fig. 15. Anti-noise capability of PBSNLR and N-PBSNLR against background voltage noise. (a) Effect of the spike trains total time duration (*Tt*). The other parameters *Fin* = 10 Hz, *Fo* = 100 Hz. (b) Effect of the firing rate of input spike trains (*Fin*). The other parameters *Tt*=400 ms, *Fo* = 100 Hz. (c) Effect of the firing rate of output spike trains (*Fo*). The other parameters *Tt*=400 ms, *Fin* = 10 Hz.

by a 2 Hz homogeneous Poisson process). The resulting plots of C_m are presented in Figs. 16 and 17.

Fig. 16 reveals that the jittering noise has a great effect on the response of the neuron trained by ReSuMe and N-ReSuMe, reflecting in the correlation C_m decreases significantly as the σ_j gradually increases. However, the correlation C_m of N-ReSuMe is significantly higher than that of ReSuMe. Therefore, the anti-noise capability against jittering noise of N-ReSuMe is more robust.

From Fig. 17, we know that the correlation C_m of PBSNLR and N-PBSNLR are both very high when the noise intensity is small. As the σ_j gradually increases, the correlation C_m of the two methods decreases. However, the extent of this decrease is much less for N-PBSNLR than for PBSNLR. That is, the anti-noise capability of the neuron trained by N-PBSNLR is better than PBSNLR.

4.3. Effects of the parameters of noise-threshold

Two of the major parameters involved in the concept of noise-threshold are η_1 and η_2. In this section, N-PBSNLR is adopted to investigate the effects of these parameters on the learning process and anti-noise capability.

4.3.1. η_1

The role of η_1 (in Eq. (10)) is to keep the membrane potential far away from the firing threshold at Nt_d. To look into the effect of η_1, a single neuron with 200 synaptic inputs is considered. The neuron is trained to output a desired spike train with a length of 400 ms. Every input and desired output spike train are Poisson spike trains with rates 10 Hz and 30 Hz, respectively. After training, the reliability of the target recall is tested under two cases: (1) testing without noise; (2) testing against background Gaussian white noise with mean 0 and $\sigma_b = 0.2$ mV. The value of η_1 varies from 0.2 to 5 mV. The experimental results are shown in Fig. 18.

The correlation C of noise free testing (i.e., learning accuracy) keeps high values with η_1 varying from 0.2 to 2 mV, which means noise-threshold has the advantage of parameter insensitivity. After that, C drops obviously. Then we investigate the correlation C of under noise. A larger η_1 results in a larger C, but when η_1 is increased above a critical value (e.g., 1 mV in our experiments), the value of C will decrease significantly. Fig. 18(b) shows that it will take more learning epochs to make the membrane potential more far away from the threshold at Nt_d.

4.3.2. η_2

We further conduct experiments to evaluate the effect of η_2 (in Eq. (11)) on the learning and anti-noise capability. In this experiment, the values of η_2 vary from 0.05 to 4 mV. After training, the reliability of the trained neuron is tested against the same noise as in Fig. 18.

As shown in Fig. 19(a), during the increase of η_2, the correlation C of noise free testing decreases. For noise testing, when η_2 varies from 0.05 to 0.25 mV, the correlation C of noise testing increases. After that, the value of C decreases gradually. From Fig. 19(b), when η_2 varies from 0.05 to 0.6 mV, N-PBSNLR needs almost the same learning epochs. However, when the value of η_2 is above 0.8 mV, there is a clear trend that N-PBSNLR needs more learning epochs with the increase of η_2.

4.4. The performance of noise-threshold in practical application

In the past decades, SNNs have been successfully applied to different applications [44–53]. In this section, we adopt a computational model [17,54,55] (see Fig. 20) to conduct several experiments on classifying patterns with noise-threshold. The model consists of three parts: encoding part, supervised learning part, and readout part. The encoding part is used to convert the input patterns into different spike trains. The learning part tunes the synaptic weights to ensure that the output part can respond to certain patterns correctly. The readout part extracts information about the stimulus from a given neural response [17].

4.4.1. Learning performance analysis of the noise-threshold

The XOR problem is a linearly nonseparable task, which is a benchmark widely used for investigating the performance of SNNs [21,55]. Thus, we use the XOR problem to investigate the performance of noise-threshold on classification tasks first.

To convert the XOR problem as a classification problem of spike trains, similar to the encoding strategy in [21,55], 0 and 1 are encoded by 100 encoding neurons, respectively. Each encoding neuron generates spike train by a Poisson process with a mean frequency *Fin*. The frequency for 0 is $Fin = 5$ Hz, and the frequency for 1 is $Fin = 10$ Hz. The spike train total time duration (Tt) is 100 ms. These input spike patterns corresponding to {1, 1} and {0, 0} are one class defined as *Class* 1 and the input spike patterns corresponding to {1,0} and {0,1} are defined as class *Class* 2. The readout part contains two neurons, with each neuron corresponding to one category. The desired outputs of the neurons corresponding to *Class* 1 and *Class* 2 are set to the spike firing times [40,80] ms and [20,60] ms, respectively. when a pattern is

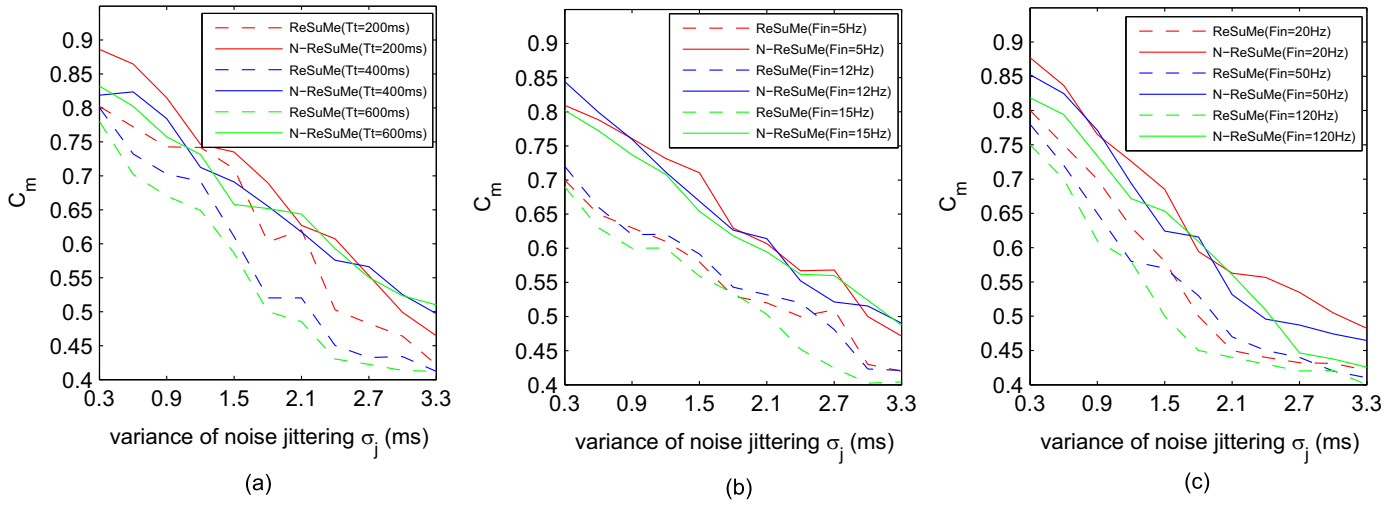


Fig. 16. Anti-noise capability of ReSuMe and N-ReSuMe against jittering noise. (a) Effect of the spike trains total time duration (Tt). The other parameters $Fin = 10$ Hz, $Fo = 100$ Hz. (b) Effect of the firing rate of input spike trains (Fin). The other parameters Tt=400 ms, $Fo = 100$ Hz. (c) Effect of the firing rate of output spike trains (Fo). The other parameters Tt=400 ms, $Fin = 10$ Hz.

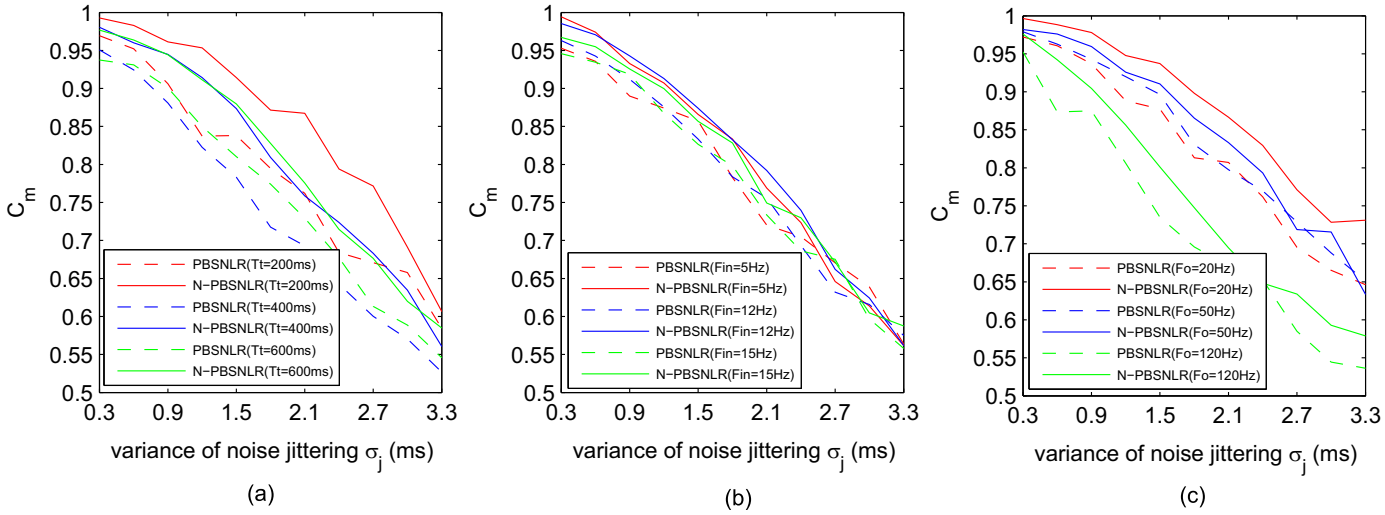


Fig. 17. Anti-noise capability of PBSNLR and N-PBSNLR against jittering noise. (a) Effect of the spike trains total time duration (*Tt*). The other parameters *Fin* = 10 Hz, *Fo* = 100 Hz. (b) Effect of the firing rate of input spike trains (*Fin*). The other parameters *Tt*=400 ms, *Fo* = 100 Hz. (c) Effect of the firing rate of output spike trains (*Fo*). The other parameters *Tt*=400 ms, *Fin* = 10 Hz.

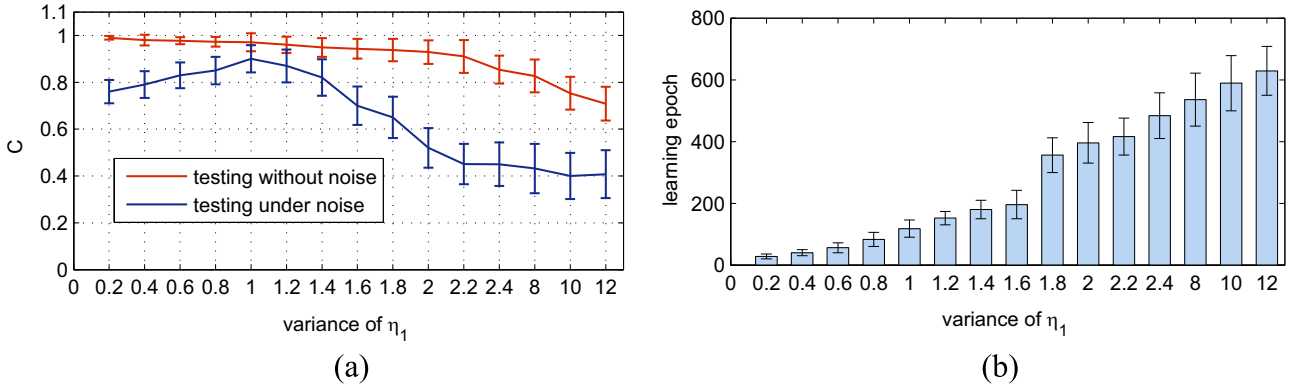


Fig. 18. Effect of η_1 on the learning and anti-noise capability. (a) Correlation C of noise testing and noise-free testing. (b) Learning epochs with different values of η_1.

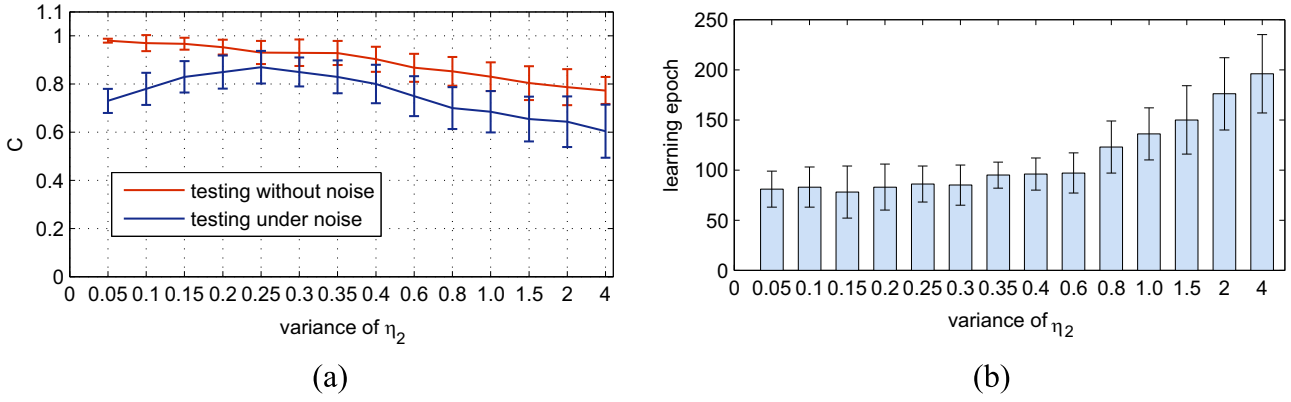


Fig. 19. Effect of η_2 on the learning and anti-noise capability. (a) Correlation C of noise testing and noise-free testing. (b) Learning epochs with different values of η_2.

present, the corresponding neuron is trained to produce target spikes, and the other neuron is trained to keep silent (not to spike). Fig. 21 (a) and (b) show the learning performance of PBSNLR and N-PBSNLR, respectively.

Fig. 21 shows both PBSNLR and N-PBSNLR can train the neuron to solve the XOR problem successfully. When patterns of {1,1} and {0, 0} present, the corresponding neuron will fire spikes at 40 ms and 80 ms. When patterns of {1, 0} and {0, 1} present, the corresponding neuron will fire spikes at 20 ms and 60 ms, respectively. Moreover, in Section 3, we have analysed that to make a neuron robust to noise, we should: (1) keep the membrane potential far away from the threshold value at Nt_d; (2) the membrane potential around t_d should be strong enough. To observe whether this is indeed the case in our experiment, we trace the membrane potential trajectories of neuron trained with different methods. Comparing of the membrane potential trajectories observed after PBSNLR and N-PBSNLR, we find that, at Nt_d, the membrane potentials of the neuron trained with N-PBSNLR (see Fig. 21(b)) are much lower than PBSNLR (see Fig. 21(a)). In addition, the membrane potential obtained by N-PBSNLR gets close to the threshold only shortly before the firing times, and the value of the membrane potential

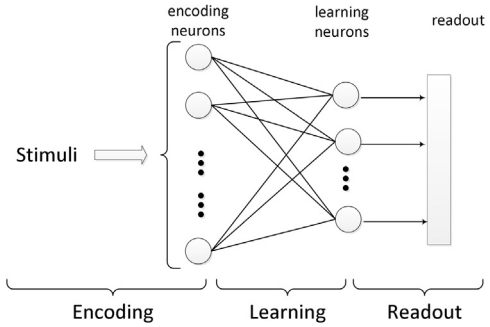


Fig. 20. General structure and information process of the SNN [54]. It contains three functional parts: encoding, learning, and readout. The encoding part is used to convert the input patterns into different spike trains. The learning part tunes the synaptic weights to ensure that the output part can respond to certain patterns correctly. The readout part extracts information about the stimulus from a given neural response.

is a little above the threshold at t_d. These observations confirm our theoretical prediction.

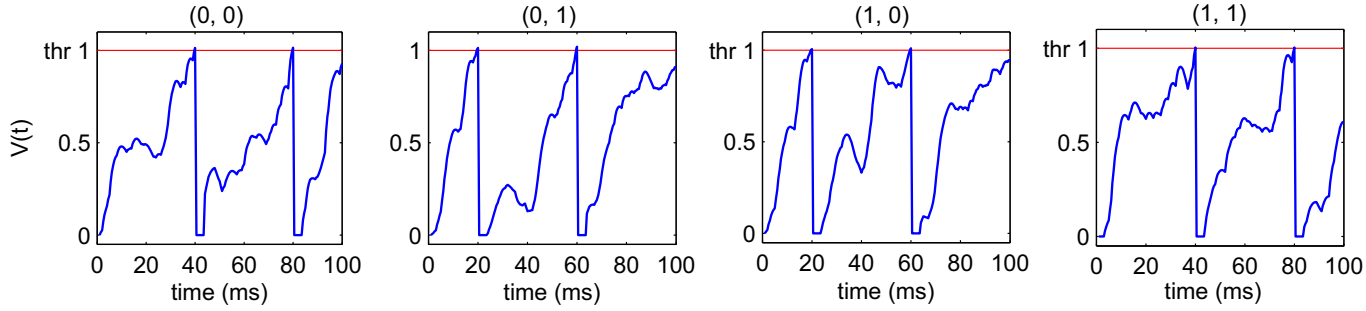
4.4.2. Optical character recognition (OCR)

To evaluate the robustness of noise-threshold in practical applications, similar to the setup in [55,56], an OCR task is considered in this experiment. A set of opticall characters of 0–9 are used. Each image has a size of 20×20 black/white pixels. The samples of OCR are illustrated in Fig. 22 (a). The phase encoding method proposed in [55,56] is used to convert the images into spatiotemporal spike patterns. Fig. 22 (b) shows an encoding result with phase coding method. The output spikes are sparsely distributed over the time window.

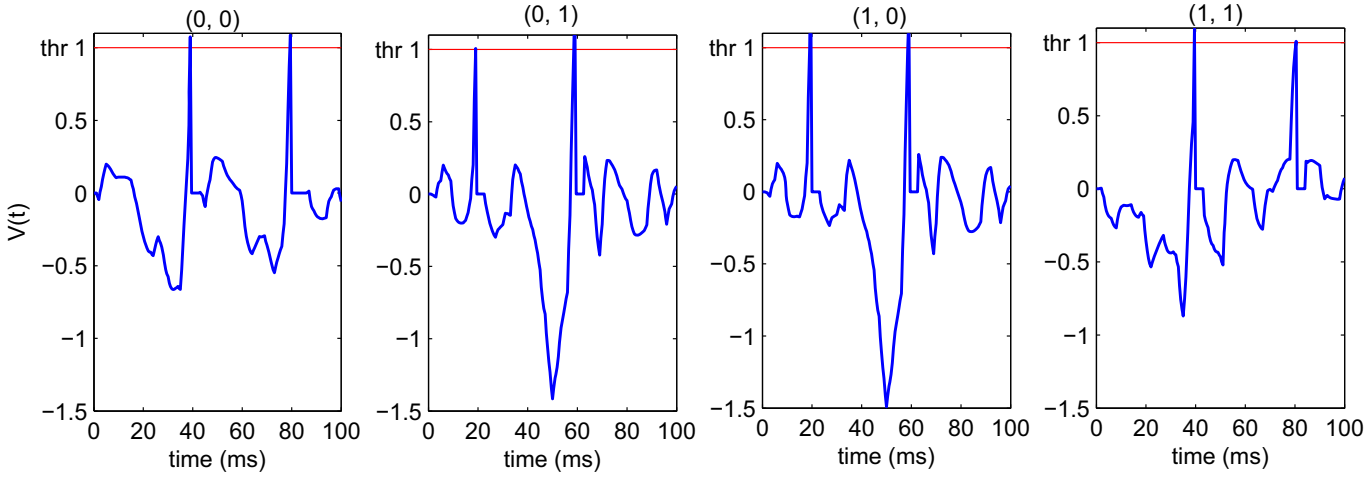
We select 10 neuron to learn to different patterns, with each learning neuron corresponding to one category. Each neuron is trained to produce a target spike train ([40, 80, 120, 160] ms) when a pattern from the assigned class is presented, and not to spike when patterns from other classes are presented [55,56]. To study the robustness, after learning, the reliability of the target recall is tested against two noise cases: (1) background noise on the membrane potential; (2) input jittering noise. The relative confidence criterion is used for making decision, where the input pattern will be decided by one of the neurons that generates the most closest spike train to the target spike train. Figs. 23 and 24 show the robust performance of different learning algorithms against background noise and jittering noise, respectively.

Fig. 23 shows that the performances of ReSuMe, PBSNLR, and DL-ReSuMe decrease with the increasing noise level. Spiking neurons trained by N-ReSuMe, N-PBSNLR, and N-DL-ReSuMe can obtain a 100% classification accuracy even when the noise level is high ($\sigma_b = 0.4$ mV). Therefore, The robustness of the supervised learning algorithms improved significantly with noise-threshold.

As can be seen from Fig. 24, the performances of all the learning rules decrease with the increasing noise level. The accuracies of ReSuMe, PBSNLR, and DL-ReSuMe drop more obviously than N-ReSuMe, N-PBSNLR, and N-DL-ReSuMe, respectively. That is, the synaptic weights obtained by N-ReSuMe, N-PBSNLR, and N-DL-ReSuMe are more robust than those of ReSuMe, PBSNLR, and DL-ReSuMe.



(a) Membrane potentials of the neurons after training with threshold (PBSNLR)



(b) Membrane potentials of the neurons after training with noise-threshold (N-PBSNLR)

Fig. 21. Performance of the noise-threshold on the XOR problem. (a) Membrane potentials of the neurons after training with threshold (PBSNLR). (b) Membrane potentials of the neurons after training with noise-threshold (N-PBSNLR).

Fig. 22. (a) OCR samples. Each image has a size of 20×20 black/white pixels. (b) Phase encoding results of a given image sample. Each dot denotes a spike.

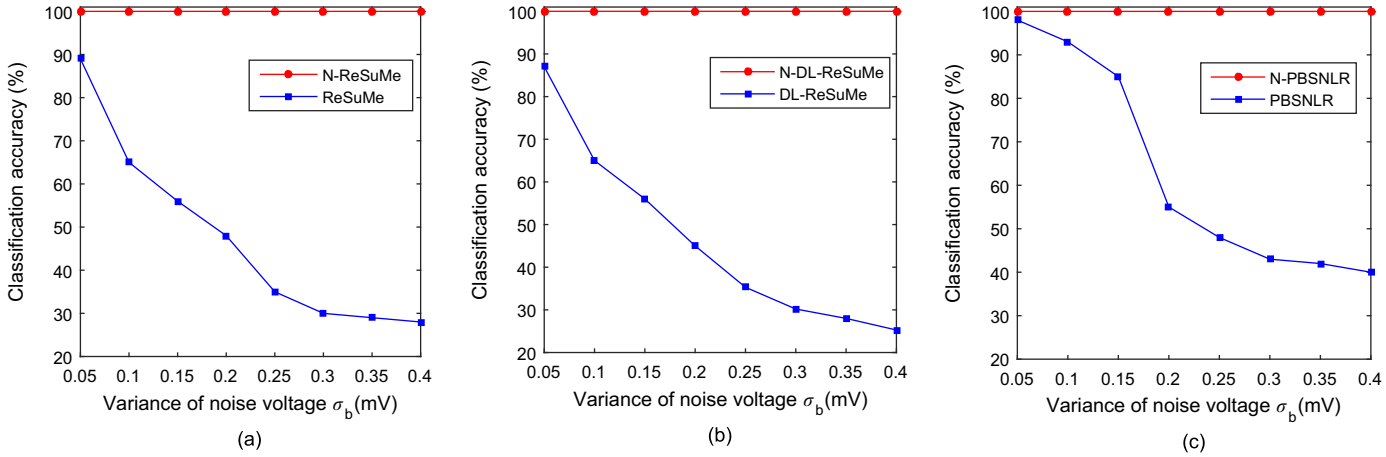


Fig. 23. Robustness of different learning algorithms against the background noise on the membrane potential. (a) Robustness of the ReSuMe and N-ReSuMe. (b) Robustness of the PBSNLR and N-PBSNLR. (c) Robustness of the DL-ReSuMe and N-DL-ReSuMe.

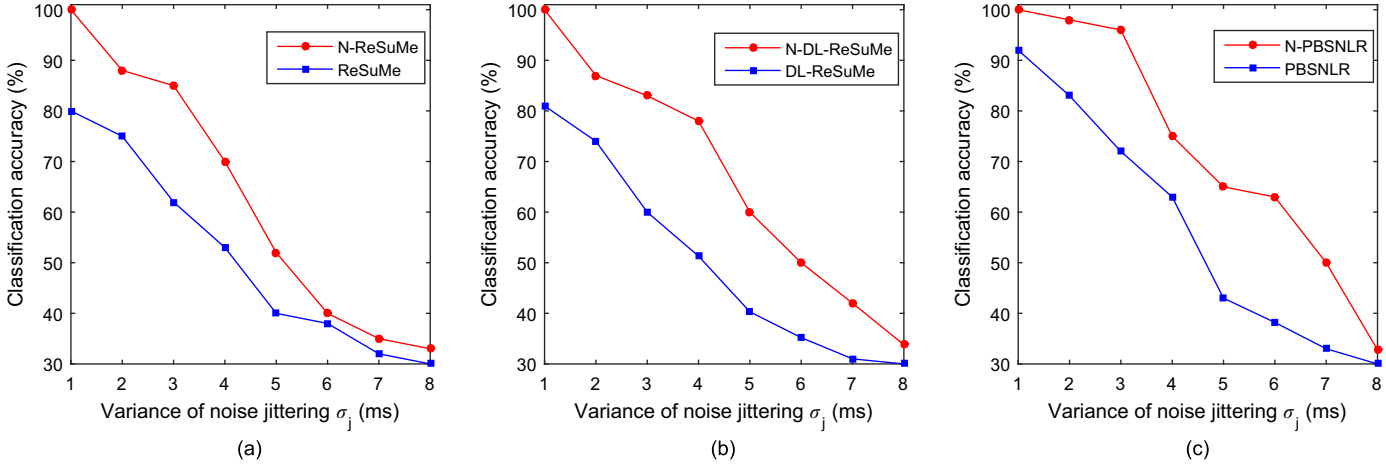


Fig. 24. Robustness of different learning algorithms against the jittering noise. (a) Robustness of the ReSuMe and N-ReSuMe. (b) Robustness of the PBSNLR and N-PBSNLR. (c) Robustness of the DL-ReSuMe and N-DL-ReSuMe.

4.4.3. Image recognition

Recently, a computational model in which visual information is encoded into precisely timed action potentials has been proposed for pattern recognition [54]. We adopt this computational model, as shown in Fig. 25, to explore the performance of noise-threshold in practical applications.

The Latency-phase encoding method proposed in [54,57] is used as the encoding mechanism to convert the images into different spike patterns and then transmit them to a spiking neural network for learning. Due to the good robust performance in previous experiments, N-PBSNLR is selected to adjust synaptic weights to make the learning neuron output desired sequences of spikes for different images. we predefine three different target spike patterns for input images. For simplicity, each desired pattern is defined as a sequence of seven spikes, [130, 180, 230, 280, 330, 380, 430] ms for image1, [150, 200, 250, 300, 350, 400, 450] ms for image2, and [170, 220, 270, 320, 370, 420, 470] ms for image3. The classification results are shown in Fig. 26.

As shown in Fig. 26, before training, the output neuron fires at random times. After several iterations of training, the learning neuron is able to reproduce different desired sequences of spikes when different input patterns are given, which means that the network could recognize all three images successfully.

Next, comparison experiments are implemented between PBSNLR and N-PBSNLR. After learning, we test the performance of the trained SNN on noisy data and noisy conditions. Noisy data is simulated by

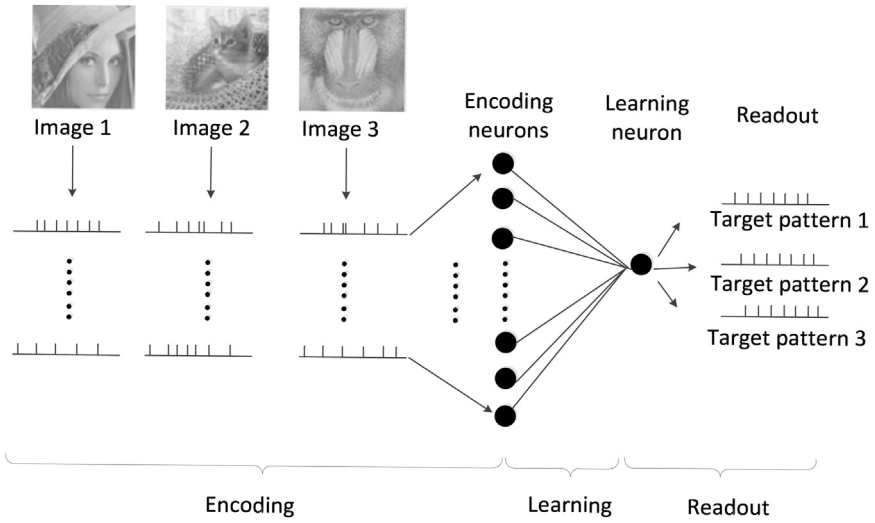


Fig. 25. General structure and information process of the SNN [54]. It contains three functional parts: encoding, learning, and readout. The encoding part is used to convert images into spike trains. After encoding, each spike train is received by one input neuron of the SNN, With a predefined target pattern for each input pattern, the SNN equipped with a supervised learning method is trained to recognize different patterns.

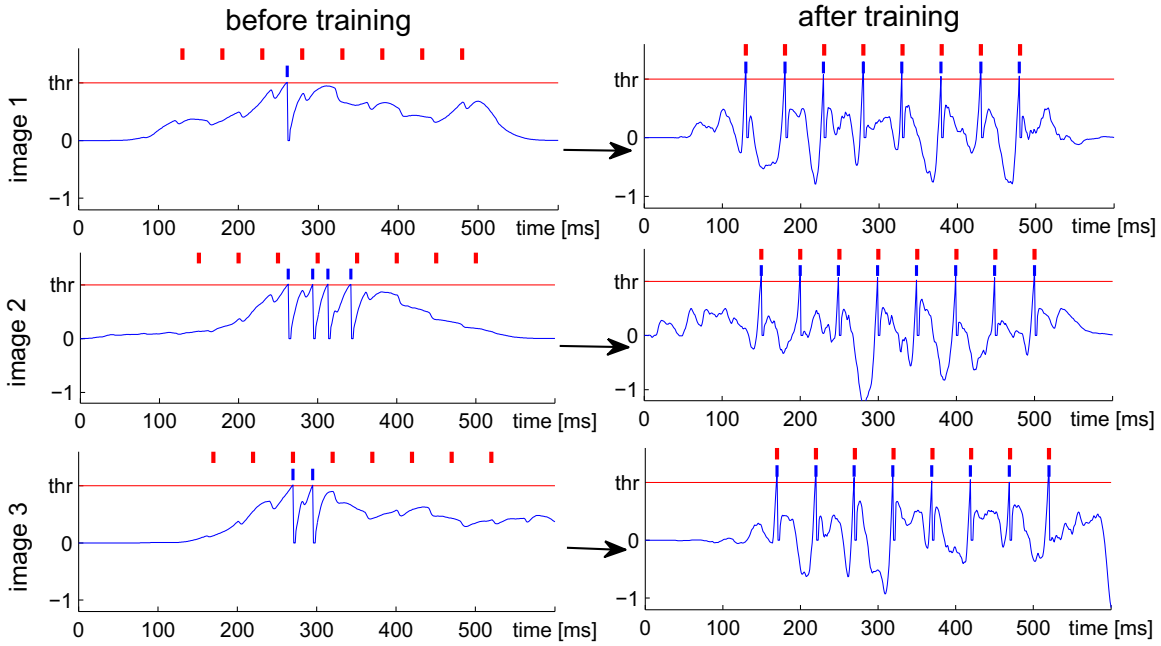


Fig. 26. Classification results of three images by N-PBSNLR. The firing threshold of spiking neurons is represented by thr. At first, the output neuron fires at random times. After training, and the trained neuron can output desired output spike trains for different images.

adding partially occluded, salt-and-pepper noise to the input images, and noisy condition is simulated by injecting Gaussian white noise voltage to the output neuron. The partially occluded noise is specified by the area of occluded n, the salt-and-pepper noise is specified by the noise intensity d, and the background noise is specified by the variance σ_b. For each kind of noise with different intensities, 50 experiments are carried out. The experimental results are shown in Fig. 27.

From Fig. 27(b), when the neuron is trained with PBSNLR, different types of noise have a great effect on the learning results. With the increase of noise intensity, the correlation C between the desired and actual output spikes decreases significantly. However, as shown in Fig. 27(c), the neuron trained with N-PBSNLR are more robust to different types of noise, and the correlation C keeps a high value even in high level of noise.

In [54], to improve the robustness of the SNN, different types and intensities of noise are added to images during training (i.e., noisy training). It should be noted that, the SNN trained under noisy condition demonstrates high robustness only to the noise during training, and it responds highly unreliably to other noise. The SNN trained with noise-threshold does not have this limitation. After deterministic training with noise-threshold, the SNN has a strong anti-noise ability to different types and intensity of noise.

5. Discussion

5.1. Comparison with the existing methods

To increase some anti-noise capability of the trained SNNs, the existing supervised learning methods adopt the strategy that train spiking neurons in the presence of noise (i.e., noisy training) [18,21,27,42,56]. However, the neuron trained under noisy conditions demonstrates relative high robustness to noise only in response to the stimuli used during the training, and the neuron responds highly unreliably to other stimuli [18]. However, the neurons trained with

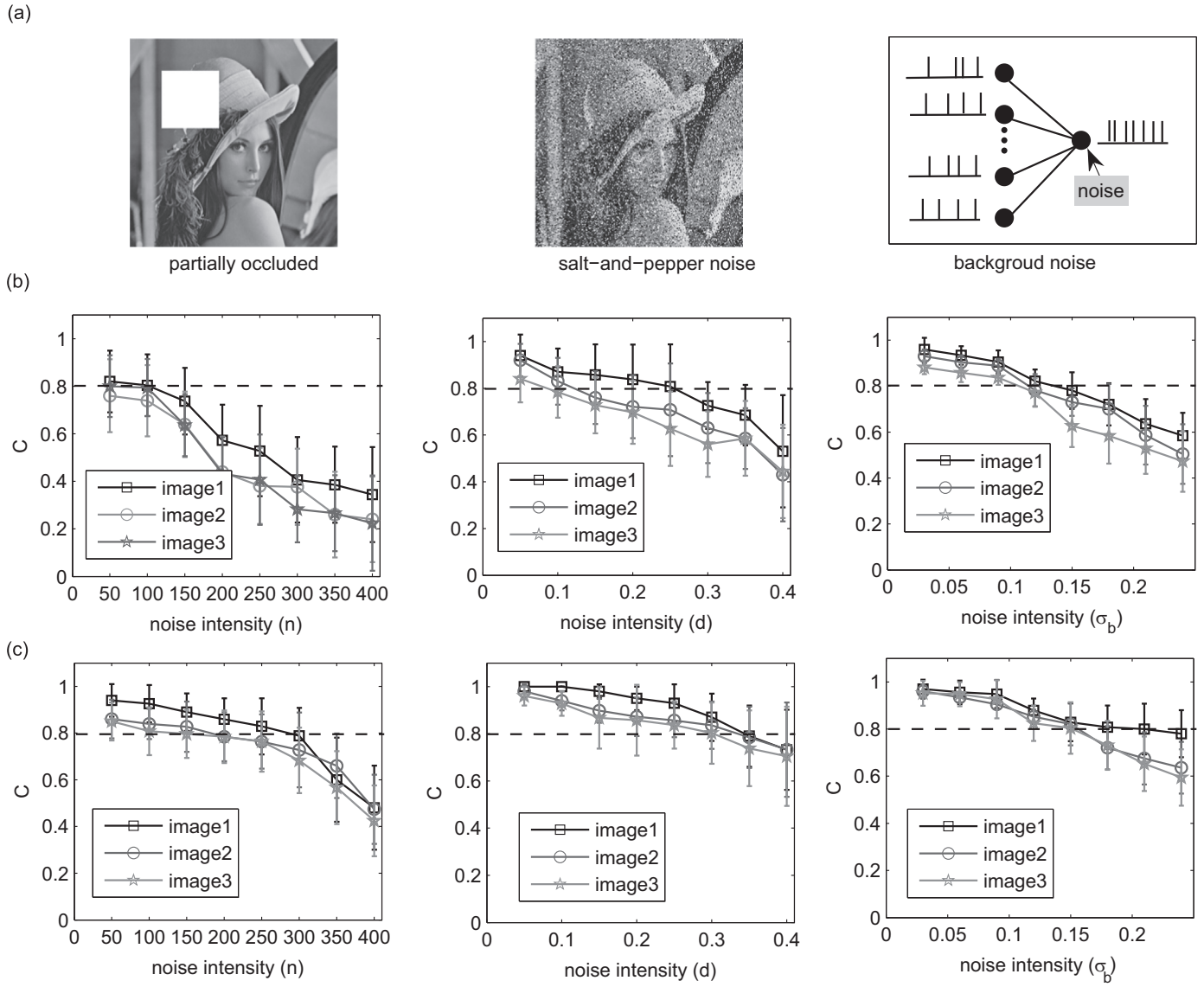


Fig. 27. Test results with different types of noise added to the input images and SNN. (a) Examples of different types of noise added to the input images and networks. (b) Response of the network trained by PBSNLR. The correlation C between output spike train and the desired spike train is used to evaluate the precision of the neural responses. (c) Response of the network trained by N-PBSNLR.

noise-threshold does not have this limitation. After deterministic training with noise-threshold, the SNN has a strong anti-noise ability to different types and intensity of noise. More importantly, noise-threshold can improve the robustness of SNN-based computational models, which is effective to deal with noisy data and can make a proper decision even under highly noisy conditions. Therefore, the proposed method will significantly promote researches and applications of spiking neural networks (SNNs).

5.2. Generality to different learning algorithms

ReSuMe and PBSNLR are selected as examples to show how noise-threshold could be combined with existing supervised learning algorithms of SNNs. If noise-threshold is limited only in ReSuMe and PBSNLR, its practicability will be largely restricted. Therefore, some generalizations of noise-threshold should be discussed. According to the properties of noise-threshold, we can summarize how to combine noise-threshold with different learning methods. For spike-driven methods, such as ReSuMe, SpikeProp, PSD, etc., we just need to change the spike generating mechanism. During the learning process, the neuron will emit an actual output spike whenever the membrane potential reaches the noise-threshold. Then, spike-driven methods can use actual and desired output spikes as signals to modify synaptic weights. The purpose of membrane potential-driven methods is to make the membrane potential above (or below) the threshold at t_d (or Nt_d) [21,28]. To create a combination with noise-threshold, we just need to change the rules to make membrane potential above (or below) noise-threshold at t_d (or Nt_d).

5.3. Different definitions of noise-threshold

In Section 3, noise-threshold is defined as Eq. (12). However, the definitions of noise-threshold can be various. The value of noise-threshold just need to below threshold at Nt_d, and a little above threshold at t_d. Fig. 28 gives some examples of different definitions of noise-threshold. In our future work, we will try to compare anti-noise capability of different definitions and select the best one for SNNs.

5.4. Anti-noise capability vs learning accuracy and efficiency

By keeping the membrane potential far away from the threshold at Nt_d and making membrane potential strong enough at t_d, experimen-

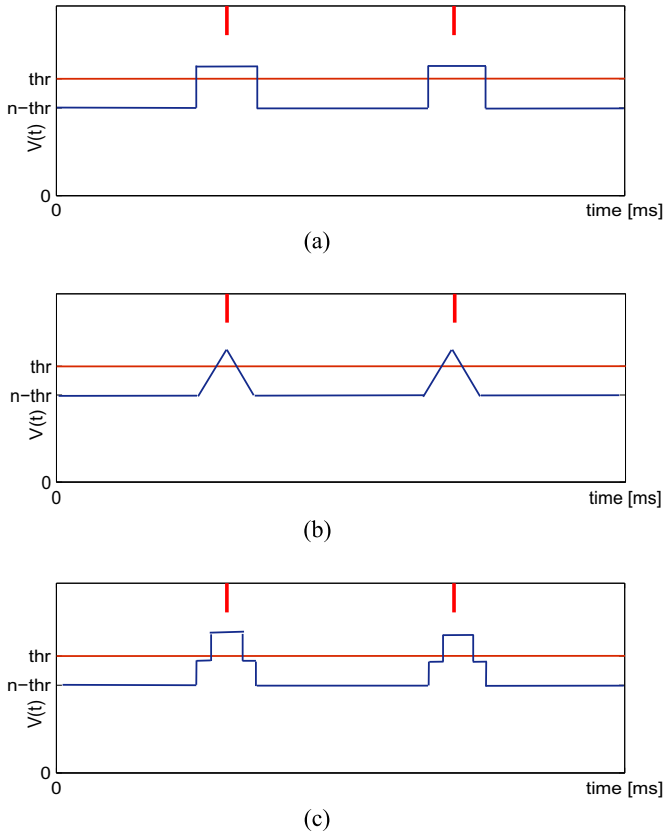


Fig. 28. Different definitions of noise-threshold. Threshold and noise-threshold are represented by thr and *n*-thr, respectively. (a) The values of noise-threshold at $\widetilde{t}_d$ and $\widetilde{N}_d$ are defined as two constants. (b) The value of noise-threshold at $\widetilde{N}_d$ is a constant, and a linear function is selected to represent the noise-threshold at $\widetilde{t}_d$. (c) The value of noise-threshold at $\widetilde{N}_d$ is a constant, and a piecewise function is selected to represent the noise-threshold at $\widetilde{t}_d$.

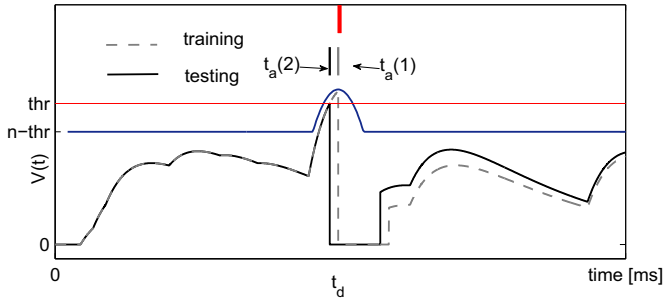


Fig. 29. A simple example to illustrate that noise-threshold gives up learning accuracy to a certain extent. The dotted line represents the membrane potential trajectory of the neuron trained with noise-threshold. The membrane potential is a little above the threshold at t_d. When we test the learning performance, as shown in the figure, the neuron will emit a spike at $t_a(2)$ instead of t_d.

tal results reveal that noise-threshold does improve the robustness of spiking neurons significantly. However, training with noise-threshold requires a little more learning epochs and gives up learning accuracy to a certain extent.

As shown in Fig. 18(b), more learning epochs are needed as η_1 increases. The reason for more learning epochs is that, combined with noise-threshold, the conditions of a successful learning is more rigorous. Noise-threshold requires the membrane potential far away from the threshold at Nt_d instead of just below it. Therefore, it will take more learning epochs to find the proper synaptic weights.

To make sure that the neuron will fire around any t_d, we set $\mathrm{n-thr} = -a[t - t_d(i)]^2 + \mathrm{thr} + \eta_2$ at $\widetilde{t}_d$. However, this setting will give up learning accuracy to a certain extent. In Fig. 19(a), with the increase of η_2, learning accuracy decreases significantly. Fig. 29 explains the reason for this phenomenon.

According to the noise-threshold, the actual firing time is very close to the desired one with a distance less than δ ($|t_d - t_a| < \delta$). Therefore, learning accuracy can be improved by adjusting the parameter δ.

5.5. About biological plausibility

The biological plausibility of noise-threshold can be embodied in the dynamic variation of firing threshold during the learning process. The biological findings in cat visual cortical and hippocampal neurons most directly confirm the variability of the firing threshold [58,59], and there are at least two separate factors that contribute to this variation: (1) fast rates of membrane potential change prior to the action potential are associated with more hyperpolarized thresholds (increased excitability); (2) the occurrence of other action potentials in the 1 s prior to any given action potential is associated with more depolarized thresholds (decreased excitability) [59].

6. Conclusion and future work

In this paper, we have put forward a dynamic firing threshold (noise-threshold) for training spiking neurons. With a combination of noise-threshold, the anti-noise capability of the existing supervised learning methods of SNNs have improved significantly, and the trained neuron is precisely and more reliably to reproduce target firing patterns even under high level noise. More importantly, noise-threshold can improve the robustness of SNNs-based computational models. Therefore, the proposed method will significantly promote researches and applications of spiking neural networks (SNNs).

For simplicity, in our experiments, we only investigate the anti-noise capability of noise-threshold defined as Eq. (12). In future work, we will try to compare anti-noise capability of different definitions and select the best one for SNNs. Another direction of our future work is to design a biological plausible SNNs-based computational model. The proposed model is expected to perform rapid and robust pattern recognition with a combination of noise-threshold.

References

[1] M.J. Berry, M. Meister, Refractoriness and neural precision, J. Neurosci. 18 (6) (1998) 2200–2211.
[2] V.J. Uzzell, E.J. Chichilnisky, Precision of spike trains in primate retinal ganglion cells, J. Neurophysiol. 92 (2) (2004) 780–789.
[3] T. Gollisch, M. Meister, Rapid neural coding in the retina with relative spike latencies, Science 319 (5866) (2008) 1108–1111.
[4] P. Reinagel, R.C. Reid, Temporal coding of visual information in the thalamus, J. Neurosci. 20 (14) (2000) 5392–5400.
[5] W. Bair, C. Koch, Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey, Neural Comput. 8 (6) (1996) 1185–1202.
[6] W. Wang, B. Subagdja, A.-H. Tan, J.A. Starzyk, Neural modeling of episodic memory: encoding retrieval, and forgetting, IEEE Trans. Neural Netw. Learn. Syst. 23 (10) (2012) 1574–1586.
[7] V.A. Nguyen, J.A. Starzyk, W.-B. Goh, D. Jachyra, NeuralNetwork structure for spatio-temporal long-term memory, IEEE Trans. Neural Netw. Learn. Syst. 23 (6) (2012) 971–983.
[8] W. Gerstner, W.M. Kistler, Spiking Neuron Models: Single Neurons, Populations, Plasticity, 1st ed., U.K. Cambridge University Press, 2002 (Aug.).
[9] S. Ghosh-Dastidar, H. Adeli, Spiking neural networks, Int. J. Neural Syst. 19 (4) (2009) 295–308.
[10] W. Maass, Networks of spiking neurons: the third generation of neural network models, Neural Netw. 10 (9) (2006) 1659–1671.
[11] W. Maass, Fast sigmoidal networks via spiking neurons, Neural Comput. 9 (2) (1997) 279–304.
[12] W. Maass, Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons. ⟨http://www.igi.tugraz.at/psfiles/90.pdf⟩.
[13] R. Kempter, W. Gerstner, J.L. Van Hemmen, Spike-based compared to rate-based Hebbian learning, Adv. Neural Inf. Process. Syst. 11 (1999) 125–131.
[14] A. Borst, F.E. Theunissen, Information theory and neural coding, Nat. Neurosci. 2 (11) (1999) 947–957.
[15] A.L. Hodgkin, A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. 117 (4) (1952)

500–544.
[16] E.M. Izhikevich, Simple model of spiking neurons, IEEE Trans. Neural Netw. 14 (6) (2003) 1569–1572.
[17] Q. Yu, H. Tang, K.C. Tan, H. Li, Rapid feedforward computation by temporal encoding and learning with spiking neurons, IEEE Trans. Neural Netw. Learn. Syst. 24 (10) (2013) 1539–1552.
[18] F. Ponulak, A. Kasinski, Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting, Neural Comput. 22 (2) (2010) 467–510.
[19] A. Taherkhani, A. Belatreche, Y. Li, et al., DL-ReSuMe: a delay learning-based remote supervised method for spiking neurons IEEE Trans. Neural Netw. Learn. Syst. 2015.
[20] A. Taherkhani, A. Belatreche, Y. Li, et al. Multi-DL-ReSuMe: Multiple neurons Delay Learning Remote Supervised Method, in Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), July, 2015, pp. 1–7.
[21] Y. Xu, X. Zeng, S. Zhong, A new supervised learning algorithm for spiking neurons, Neural Comput. 25 (6) (2013) 1472–1511.
[22] R.M. Memmesheimer, R. Rubin, B.P., et al. Learning precisely timed spikes. Neuron, vol. 82, no. 4, 2014, pp. 925–938
[23] C. Albers, M. Westkott, K. Pawelzik, Learning of Precise Spike Times with Homeostatic Membrane Potential Dependent Synaptic Plasticity. arXiv preprint arXiv:1407.6525.
[24] S.M. Bohte, J.N. Kok, H.La. Poutre, Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing 48 (1) (2002) 17–37.
[25] R. Gtig, H. Sompolinsky, The tempotron: a neuron that learns spike timing-based decisions, Nat. Neurosci. 9 (3) (2006) 420–428.
[26] R.V. Florian, The chronotron: a neuron that learns to fire temporally precise spike patterns, PloS One 7 (8) (2013).
[27] A. Mohemmed, S. Schliebs, S. Matsuda, N. Kasabov, Span: spike pattern association neuron for learning spatio-temporal spike patterns, Int. J. Neural Syst. 22 (4) (2012).
[28] M. Zhang, Q. Hong, J. Li, X. Xie, A new supervised learning algorithm for spiking neurons., in: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, 2015, vol. 1.
[29] A. Taherkhani, A. Belatreche, Y. Li, et al., A new biologically plausible supervised learning method for spiking neurons, Proc. ESANN, 2014.
[30] A. Manwani, C. Koch, Detecting and estimating signals in noisy cable structures, I: neuronal noise sources, Neural Comput. 11 (1999) 1797–1829.
[31] E. Schneidman, B. Freedman, I. Segev, Ion channel stochasticity may be critical in determining the reliability and precision of spike timing, Neural Comput. 10 (1998) 1679–1703.
[32] C. van Vreeswijk, H. Sompolinsky, Chaos in neuronal networks with balanced excitatory and inhibitory activity, Science 274 (1996) 1724–1726.
[33] N. Brunel, V. Hakim, Fast global oscillations in networks of integrate-and-fire neurons with low firing rates, Neural Comput. 11 (1999) 1621–1671.
[34] F. Rieke, Spikes: Exploring the Neural Code, MIT Press.
[35] E. Schneidman, Noise and Information in Neural Codes (Unpublished doctoral dissertation), Hebrew University.
[36] M.C. van Rossum, B.J.O. Brien, R.G. Smith, Effects of noise on the spike timing precision of retinal ganglion cells, J. Neurophysiol. 89 (2003) 2406–2419.
[37] S. Fusi, P. Del Giudice, D.J. Amit, Neurophysiology of a VLSI spiking neural network: LANN21. IJCNN 2000, vol. 3, 2000 pp. 121–126.
[38] E. Chicca, S. Fusi, Stochastic synaptic plasticity in deterministic a VLSI networks of spiking neurons, in: Proceedings of the World Congress on Neuroinformatics, 2000.
[39] Z.F. Mainen, T.J. Sejnowski, Reliability of spike timing in neocortical neurons, Science (1995) 1503–1506.
[40] T. Shmiel, R. Drori, O. Shmiel, Y. Ben-Shaul, Z. Nadasdy, M. Shemesh, Neurons of the cerebral cortex exhibit precise interspike timing in correspondence to behavior, Proc. Natl. Acad. Sci. USA (2005) 18655–18657.
[41] M.A. Montemurro, S. Panzeri, M. Maravall, A. Alenda, M.R. Bale, M. Brambilla, Role of precise spike timing in coding of dynamic vibrissa stimuli in somatosensory thalamus, J. Neurophysiol. (2007) 1871–1882.
[42] F. Ponulak, Supervised Learning in Spiking Neural Networks with ReSuMe Method (Ph.D. thesis), Poznasn University of Technology, Poland, 2006.
[43] S. Schreiber, J.M. Fellous, D. Whitmer, P. Tiesinga, T.J. Sejnowski, A new correlation-based measure of spike timing reliability, Neurocomputing 52 (2003) 925–931.
[44] D. Kerr, T.M. McGinnity, S. Coleman, M. Clogenson, A biologically inspired spiking model of visual processing for image feature detection, Neurocomputing 158 (2015) 268–280.
[45] Z. Wang, L. Guo, M. Adjouadi, Wavelet decomposition and phase encoding of temporal signals using spiking neurons, Neurocomputing 173 (2016) 1203–1210.
[46] S. Wei, H. Qu, Automatic image segmentation based on PCNN with adaptive threshold time constant, Neurocomputing 74 (9) (2011) 1485–1491.
[47] S. Dora, K. Subramanian, S. Suresh, N. Sundararajan, Development of a self-regulating evolving spiking neural network for classification problem, Neurocomputing 171 (2016) 1216–1229.
[48] H. Rostro-Gonzalez, et al., A CPG system based on spiking neurons for hexapod robot locomotion, Neurocomputing 170 (2015) 47–54.
[49] H. Qu, S.X. Yang, et al., Real-time robot path planning based on a modified pulse-coupled neural network model, IEEE Trans. Neural Netw. 20 (11) (2009) 1724–1739.
[50] H. Rostro-Gonzalez, et al., A CPG system based on spiking neurons for hexapod robot locomotion, Neurocomputing, 2015.
[51] D. Kerr, T.M. McGinnity, S.C. Marine, A biologically inspired spiking model of visual processing for image feature detection, Neurocomputing 158 (2015) 268–280.
[52] H. Qu, X.R. Xie, Y.S. Liu, M.L. Zhang, L. Lu, Improved perception-based spiking neuron learning rule for real-time user authentication, Neurocomputing 151 (2015) 310–318.
[53] J. Wang, A. Belatreche, L. Maguire, et al., An online supervised learning method for spiking neural networks with adaptive structure, Neurocomputing 144 (2014) 526–536.
[54] J. Hu, H. Tang, K.C. Tan, H. Li, L. Shi, A spike-timing-based integrated model for pattern recognition, Neural Comput. 25 (2) (2013) 450–472.
[55] Q. Yu, R. Yan, H. Tang, K.C. Tan, H. Li, A spiking neural network system for robust sequence recognition, IEEE Trans. Neural Netw. Learn. Syst. 2015.
[56] Q. Yu, H. Tang, K.C. Tan, H. Li, Precise-spike-driven synaptic plasticity: learning hetero-association of spatiotemporal spike patterns, Plos One, 2013
[57] Z. Nadasdy, Information encoding and reconstruction from the phase of action potentials, Front. Syst. Neurosci. 2009.
[58] R. Azouz, C.M. Gray, Dynamic spike threshold reveals a mechanism for synaptic coincidence detection in cortical neurons in vivo, Proc. Natl. Acad. Sci. 97 (2000) 8110–8115.
[59] D.A. Henze, G. Buzsaki, Action potential threshold of hippocampal pyramidal cells in vivo is increased by recent spiking activity, Neuroscience 105 (1) (2001) 121–130.

Malu Zhang is a current Ph.D. student in Department of computer science and engineering, University of Electronic Science and Technology of China. His current research interests relate to Neural Networks, Machine Learning and Big Data.

Hong Qu received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, in 2006. From 2007 to 2008, he was a Postdoctoral Fellow at the Advanced Robotics and Intelligent Systems Lab, School of Engineering, University of Guelph, Guelph, ON, Canada. From 2014 to 2015, he work as a visiting scholar in the Potsdam Institute for Climate Impact Research(PIK), 14473 Potsdam, Germany, and in Humboldt University of Berlin. Currently, he is a professor in Computational Intelligence Laboratory, School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include Neural Networks, Machine Learning and Big Data.

Xiurui Xie is a current Ph.D. student in Department of computer science and engineering, University of Electronic Science and Technology of China. Her current research interests relate to Neural Networks, Machine Learning and Big Data.

Jurgen Kurths received the B.S. degree in mathematics from the University of Rostock; the Ph.D. degree from the Academy of Sciences of the German Democratic Republic in 1983; the Honorary degree from N.I. Lobachevsky State University of Nizhny Novgorod in 2008; and the Honorary degree from Saratov State University in 2012.

From 1994 to 2008, he was a Full Professor with the University of Potsdam, Potsdam, Germany. Since 2008, he has been a Professor of nonlinear dynamics with Humboldt University, and the Chair of the Research Domain Transdisciplinary Concepts with the Potsdam Institute for Climate Impact Research, Germany. Since 2009, he has been the Sixth-Century Chair of Aberdeen University, Aberdeen, U.K. He is the author of more than 480 papers, which are cited more than 25000 times (H-factor: 71). His main research interests include synchronization, complex networks, time series analysis, and their applications.

Dr. Kurths is a Fellow of the American Physical Society and a Member of the Academia Europea and Macedonian Academy of Sciences and Arts. He is an Editor for PLos ONE, Philosophical Transactions of the Royal Society A, CHAOS, etc. He received the Alexander von Humboldt Research Award from the Council of Scientific and International Research (India).