Applied Intelligence (2023) 53:19466–19490
https://doi.org/10.1007/s10489-023-04553-0

Research Progress of spiking neural network in image classification: a review

Li-Ye Niu [1] · Ying Wei [1,2] · Wen-Bo Liu [1] · Jun-Yu Long [1] · Tian-hao Xue [1]

Accepted: 1 March 2023 / Published online: 8 March 2023

Abstract
Spiking neural network (SNN) is a new generation of artificial neural networks (ANNs), which is more analogous with the brain. It has been widely considered with neural computing and brain-like intelligence. SNN is a sparse trigger event-driven model, and it has the characteristics of hardware friendliness and energy saving. SNN is more suitable for hardware implementation and rapid information processing. SNN is also a powerful method for deep learning (DL) to study brain-like computing. In this paper, the common SNN learning and training methods in the field of image classification are reviewed. In detail, we examine the SNN algorithms based on synaptic plasticity, approximate backpropagation (BP), and ANN to SNN. This paper comprehensively introduces and tracks the latest progress of SNN. On this basis, we also analyze and discuss the challenges and opportunities it faces. Finally, this paper prospects for the future development of SNN in the aspects of the biological mechanism, network training and design, computing platform, and interdisciplinary communication. This review can provide a reference for the research of SNN to promote its application in complex tasks.

Keywords Spiking neural network · Artificial neural network · Deep learning · Network training · Biological mechanism

1 Introduction

The human brain relies on neurons to form a complex nervous system to process information [1]. Inspired by the brain, artificial neural network (ANN) is widely used in the field of artificial intelligence (AI) [2]. Although ANN has made outstanding achievements in various tasks, its principle is still far from that of brain information processing [3]. In the early 1950s, Hodgkin et al. began to study the electrochemical characteristics of neurons and described the spike firing behavior in the form of eqs. [4]. By the late 1980s, biologists found that synchronized oscillations in the cat's visual cortex attracted the attention of neuroscience [5]. With the gradual clarification of the brain information processing mechanism, the spike neural network (SNN) inspired by biological neurons has been proposed [6]. The mechanism of SNN processing perceptual information is close to the human brain, which is called the third-generation neural network.

Brain neurons rely on action potentials to process stimulus information [7]. ANN can only process the spatial dimension information of the image and its internal data process analog information, so its biological rationality has been questioned [8]. SNN uses more biologically interpretable spiking neurons as the basic unit and uses the discrete spike train to represent and process images [9]. The spike train integrates many aspects of information, such as time, space, frequency, and phase [10]. SNN processing of discrete spike information prevents its training from being unified [11].

In the past 40 years, biologists studying brain mechanisms have made extensive efforts to develop a general SNN learning and training model [12, 13]. The learning of SNN and the correction of synaptic weight depend on synaptic dynamics [14]. The researchers found that the sustained response of neurons caused by synaptic activity is manifested in the change of synaptic strength, so they proposed spike-time dependent plasticity (STDP) [15]. STDP is the primary mechanism of biological brain learning and memory [16]. The biological STDP based on spike control is compatible with the spiking neuron model. In 2000, Rossum et al. proposed an unsupervised Hebb mechanism based on the Hebbian rule, and the learning stability of the STDP has been dramatically

✉ Ying Wei
weiying@ise.neu.edu.cn

1 College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

2 Information Technology R&D Innovation Center of Peking University, Shaoxing, China

improved [17]. In 2000 and 2001, Song et al. used SNN based on the Hebbian rule to simulate the brain's nervous system [18]. The performance of the above training mechanism can reach the same level as ANN in shallow networks, but its performance is not satisfactory in deep SNN [19]. SNN cannot be trained by the error backpropagation (BP) method [9]. Therefore, the approximate error BP algorithm is proposed, which also shows excellent performance. The most effective way to achieve deep SNN is to convert ANN into SNN [20]. However, this method will cause a loss of network performance [21]. Based on this, developing an efficient and general SNN algorithm for unstructured perceptual information such as image processing (IP) is of great significance to brain-like intelligence.

Compared with ANN, SNN deployed on a neuromorphological chip will have stronger information processing ability [11]. SNN is an event-driven and sparsely triggered network. Its calculation and energy consumption mainly focus on the moment when neurons emit spikes [22]. Unlike ANN, which works continuously at a high level, it effectively saves the energy consumption of neurons in an inactive state [23]. On the other hand, SNN can be simulated on a special brain-like chip [24], such as IBM's TureNorth chip and SpiNnaker super brain-computer. SNN is still developing rapidly, and it has more significant application potential for mobile devices [25]. SNN has achieved fruitful research results in IP, and its applications involve informatics [20], medicine [26], social economy [21], and other fields [27], as shown in Fig. 1. ANN and SNN differ greatly in terms of network composition and information processing methods. Table 1 compares the advantages and disadvantages of the two models. Unlike ANN, SNN lacks a complete theoretical framework and well-established training methods. Moreover, the research on neuromorphological chips is also in its infancy. This limits the network depth of SNN and its application in complex tasks. This is also the main reason why SNN is mainly used for simple IP at present [28].

On this basis, this paper summarizes the research work of SNN in the field of image classification. In the first section, we introduce the neural network development, analyze the background of SNN and its information processing mechanism. In the second section, the biological background and composition of SNN are presented, including spiking neuron models, training algorithms, synaptic plasticity mechanisms, and information coding. In the third section of the paper, we provide a detailed review of SNN work in the field of image classification according to different model training methods. Then, we introduce other new SNNs and their applications. Based on the above analysis, the paper finally discusses the challenges and opportunities faced by the current development of SNN.

2 Spiking neural network

In this section, we introduce the composition of SNN, including the spiking neuron model, information processing mechanism and coding method.

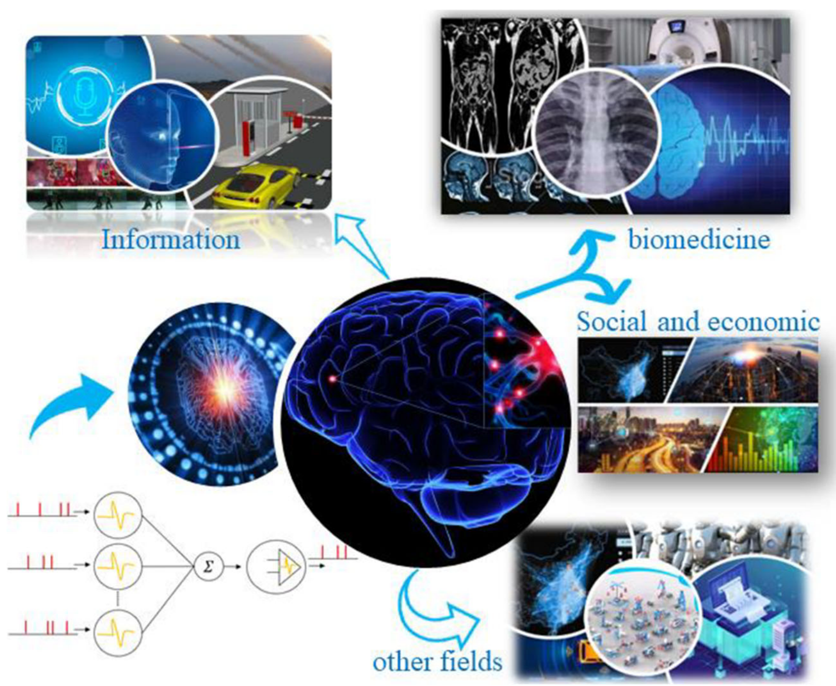


Fig. 1 Application fields of spiking neural network

Table 1 The advantages and disadvantages of ANN and SNN are compared

Model Category	Spiking Neural Network (SNN)	Artificial Neural Network (ANN)
Advantages	• Event driven, low power consumption, high efficiency. • It can run on a special neuromorphological chip.	• Good performance and high precision. • It is suitable for a variety of complex computer tasks.
Disadvantages	• Low accuracy, suitable for simple tasks. • The theory of SNN is not yet mature. Deep SNN model training is difficult.	• In addition to GPU, there is a lack of dedicated acceleration chips. • High power consumption and poor real-time performance require a lot of computing resources.

2.1 Neuron

The neuron is the basic unit of the biological nervous system [29]. The spiking neuron processes the input information and converts the spike into the membrane potential [30]. The essence of the spiking neuron model is to equivalent neurons to *RC* circuits and simulate neurons through capacitance *C* and resistance *R* [31], as shown in Fig. 2a. The spiking neuron is surrounded by the cell membrane, which can be regarded as an insulator. When the external current $I(t)$ is injected into the neuron, the extra charge charges the cell membrane. At this time, the cell membrane is equivalent to a capacitor *C*. Since the cell membrane insulator is not perfect, the electric charge slowly seeps from the cell membrane. Therefore, the cell membrane also has a limited leakage resistance *R*. $V(t)$ and u_{rest} represent the current membrane potential and resting potential of neurons respectively. According to the law of conservation of current, the driving current $I(t)$ can be divided into two components, I_R and I_C, so $I(t) = I_R + I_C$. I_R represents the current flowing through the resistance *R*. I_C represents the charging current of capacitor *C*.

The membrane potential accumulation and spike emission of spiking neurons are shown in Fig. 2b. When neurons receive spike information, their membrane potential will increase. When the accumulation of membrane potential reaches the threshold V_{thr}, neurons will emit spike signals in the form of the action potential, and the corresponding membrane potential will be reset [32]. After the reset of spiking neurons, their membrane potential is in a resting state.

Next, we introduce two typical spiking neuron models, including Integrate-and-Fire (IF) model and Leaky Integrate-and-Fire (LIF) model.

2.1.1 Integrate-and-fire (IF) model

The leakage of current is not considered in the process of neuronal membrane potential accumulation of the IF model. The change of neuron membrane potential can be described by the first-order differential equation [33]. Equation (1) gives the relationship between membrane potential $V(t)$ and current $I(t)$.

$$\frac{dV}{dt} = \frac{1}{C} I(t). \tag{1}$$

When there is a current input to the model, the voltage at the two ends of the membrane capacitor accumulates over time and exceeds the threshold, and the neuron emits a spike. When there is no external current input, the neuronal membrane potential will always maintain its original state and will not change.

2.1.2 Leaky integrate-and-fire (LIF) model

When biological neurons do not have any input, their membrane potential will decay with time. Therefore, the LIF model introduces a leakage term to simulate the ion diffusion effect of cells [34]. Compared with the IF model, the leakage term is added to the first-order differential equation of the model. Its membrane potential $V(t)$ changes with time as shown in Eq. (2).

$$\tau_m \frac{dV}{dt} = -(V(t) - u_{rest}) + RI(t), \tag{2}$$

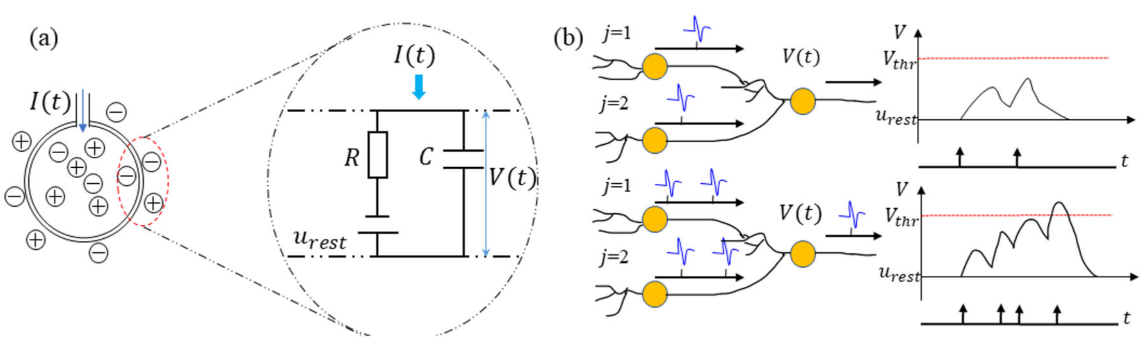


Fig. 2 **a** The equivalent model of biological neurons. **b** The membrane potential accumulation and spike emission of spiking neurons

where time constant τ_m = RC and u_{rest} is the resting potential (u_{rest} = 0). When the membrane potential is less than the threshold, the neuron will not emit spikes, and its membrane potential will change continuously according to the time constant τ_m decreased to u_{rest}. This model is more consistent with the change mechanism of neuronal membrane potential in the biological system. In addition to IF and LIF models, there are Izhikevich [15] and Hodgkin-Huxley [4] models with many parameters. The spiking neuron is the abstraction and approximation of biological neurons. Different models have different bionics and computational complexity [35].

2.1.3 SNN simulation of spiking neurons

The dynamic equations (Eq. (1) and Eq. (2)) of IF and LIF neurons are nonlinear differential equations, and the state variables of neurons cannot be directly obtained in SNN simulation. Thus, the dynamical equations need to be equivalent to the form that can be used for network training to approximate numerical simulations. The equivalent forms of Eq. (1) and Eq. (2) are shown in Eq. (3).

$$V_i^l(t) = \begin{cases} V_i^l(t-1) + \sum_{j=1}^{n(l-1)} \omega_{ij}^l s_j^{l-1}(t) & IF \quad model \\ \tau_{decay} * V_i^l(t-1) + \sum_{j=1}^{n(l-1)} \omega_{ij}^l s_j^{l-1}(t) & LIF \ model \end{cases}, \tag{3}$$

where $V_i^l(t)$ represents the membrane potential of neuron i of layer l network at time t. $n(l-1)$ represents the number of neurons in layer $l-1$ of the network. ω_{ij}^l represents the weight between neuron i of layer l and neuron j of the previous layer. $s_j^{l-1}(t)$ represents the spike state of layer $l-1$ neuron j at time t (there is a spike $s_j^{l-1}(t)$ =1 at the current time of the neuron, otherwise, $s_j^{l-1}(t)$ =0). The comparison equation shows that $I(t) = \sum_{j=1}^{n(l-1)} \omega_{ij}^l s_j^{l-1}(t)$ in the dynamic equations of IF and LIF. The membrane potential calculation of the LIF model is one more attenuation factor τ_{decay} than that of the IF model. τ_{decay} is a constant describing the attenuation rate of membrane potential (τ_{decay} here represents the leakage of the membrane potential of the LIF model).

2.1.4 Reset mechanism of spiking neurons

The neuron emits a spike and then the neuron performs a reset operation. There are two kinds of neuronal reset modes: hard-reset and soft-reset, as shown in Eq. (4).

$$V(t) = \begin{cases} (V(t-1) + I(t)) * (1 - S(t)) & \text{hard-reset} \\ (V(t-1) + I(t)) - V_{thr} * S(t) & \text{soft-reset} \end{cases}, \tag{4}$$

where $V(t)$ and $I(t)$ represent the membrane potential and input of neurons at time t, respectively. V_{thr} represents the threshold of neurons. $S(t)$ represents whether the neuron has a spike at time t (when $S(t)$ = 1 means that there is a spike, and $S(t)$ = 0 means that the neuron does not emit a spike). It can be seen from the analysis of Eq. (3) that when a neuron performs the hard reset operation, the membrane potential of the neuron will be reset to 0 (generally 0 is the resting potential); When a neuron performs the soft reset operation, a threshold value is subtracted from the membrane potential.

2.1.5 Threshold analysis of spiking neurons

The threshold V_{thr} of a spiking neuron is an important parameter that affects its spike firing rate F_{sfr}. The corresponding relationship between F_{sfr} and V_{thr} is shown in Fig. 3. When the threshold is very small, the F_{sfr} will be oversaturated (Phase 1). In this case, spiking neurons are very sensitive to input information. When the threshold is large, the spiking neuron will not emit a spike (Phase 3). In this case, the spiking neuron loses the ability to distinguish the input information. The setting of the threshold and spike firing rate should reach a balance (Phase 2), which is the key to ensure SNN performance [36]. The threshold of the model needs to be set by combining the coding method and the training algorithm. For example, in the algorithm for converting ANN to SNN, the threshold is generally set to 1 if the model weights are normalized. If the model weight is not normalized, the threshold value is generally set to the maximum value of neuron activation value [37].

2.2 Learning mechanism

The basis of nervous system learning is synaptic plasticity, which plays an important role in SNN [38]. The STDP mechanism is a local learning algorithm [39]. The relative time difference between presynaptic and postsynaptic spikes determines the direction and size of synaptic weight changes [40]. We assume that the presynaptic neuron i emits a spike at time t_i^f and the postsynaptic neuron j emits a spike at time t_j^f. Then, the relative time difference $s = t_j^f - t_i^f$. The relative timing of

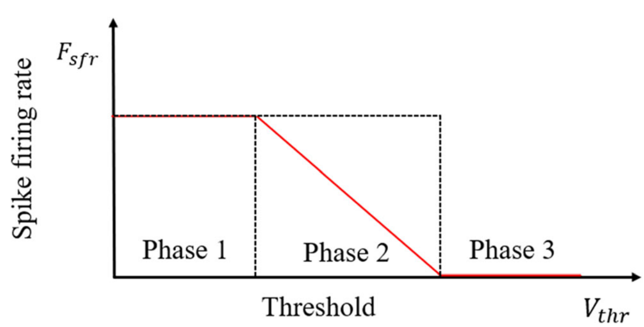


Fig. 3 The corresponding relationship between F_{sfr} and V_{thr}

spikes emitted by neurons before and after synapse will induce different synaptic change processes [41], and the learning window of synaptic weight is shown in Fig. 4. When $s > 0$, the synaptic weight increases. The smaller the s, the more the weight of the synapse. With the decrease of s, the change of synaptic weight will decrease accordingly, and finally tend to 0. When $s < 0$, the synaptic weight decreases.

As a classical plasticity mechanism, the expression of synaptic weight $W(s)^{STDP}$ of STDP is shown in Eq. (5).

$$W(s)^{STDP} = \begin{cases} +A_+ e^{-\frac{s}{\tau_+}}, s \geq 0 \\ -A_- e^{-\frac{s}{\tau_-}}, s < 0 \end{cases}, \tag{5}$$

where A_+ and A_- are both positive numbers, indicating the maximum value of synaptic changes. τ_+ and τ_- are the time constant of boost and decay, respectively. At present, STDP has many expressions of mathematical functions, and its basic form is similar to Eq. (5). According to the difference of $A_\pm$, STDP is divided into *additive STDP* and *multiplicative STDP* [42]. According to the different spike pairing modes, STDP can be divided into *all-to-all scheme* and *nearest-neighbor scheme* [43]. In SNN research, the above mechanism is mainly applied to the direct training of networks.

2.3 Spiking neural network coding

External stimulation signals can be processed by the nervous system only when they are encoded into the form of action potentials by specific neurons [44]. The selection of the information coding method is a prerequisite for studying biological perception behavior [45]. So far, researchers do not have a complete and accurate understanding of the information coding [46]. The most common coding methods include spike frequency [47], spike time (the coding process is shown in Fig. 5a and b), and neuron population coding (Gaussian coding is shown in Fig. 5c) [48].

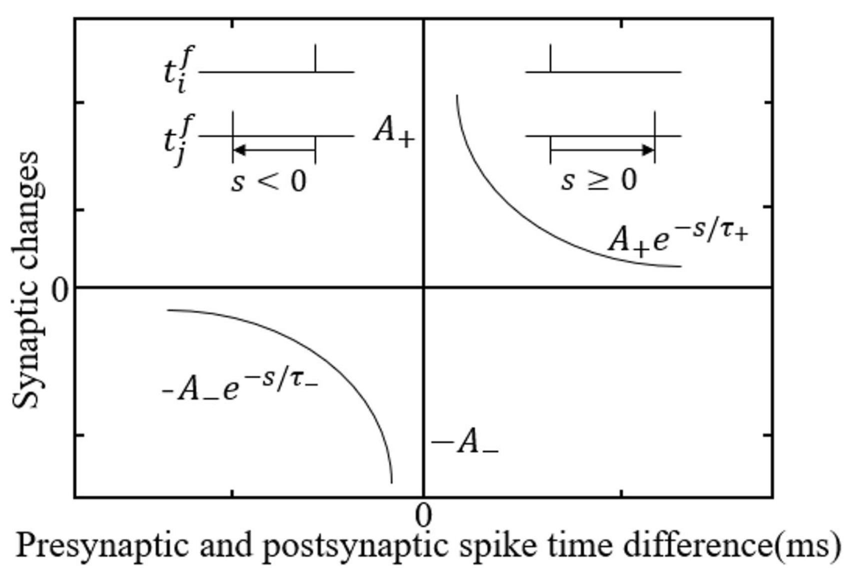
Poisson coding is a rate coding method. A Poisson event generation process is used to generate the input spike train [49]. In addition to Poisson coding, spike frequency-based coding also includes frequency coding based on spike count, spike density, and group activity [50]. First-spike time coding is a common method of spike precise timing coding [51]. This coding method defaults that neurons produce only one spike. The earlier the spike generation time is, the stronger the stimulation is. The later the spike generation time is, the weaker the stimulation signal is [52].

The encoding, spike rate, and time window of the input image affect the performance of the network. A large number of coding spikes will cause redundancy of information and waste computing resources [53]. A small number of spikes and a small-time window will cause the loss of information. Therefore, different networks need to select appropriate coding methods for specific tasks [54].

3 SNN learning algorithm

In sections 1 and 2, we introduce the composition, architecture, and development of SNN. The development process of SNN can be roughly divided into the following stages [55], as shown in Fig. 6. The development of SNN originates from the Hebb rule and the Hodgkin-Huxley [4] model. With the research of neuronal synaptic mechanisms and the introduction of neural computing, STDP has been proposed to simulate biological systems [56]. STDP combined with a new biological mechanism

Fig. 4 Synaptic weight change in learning window



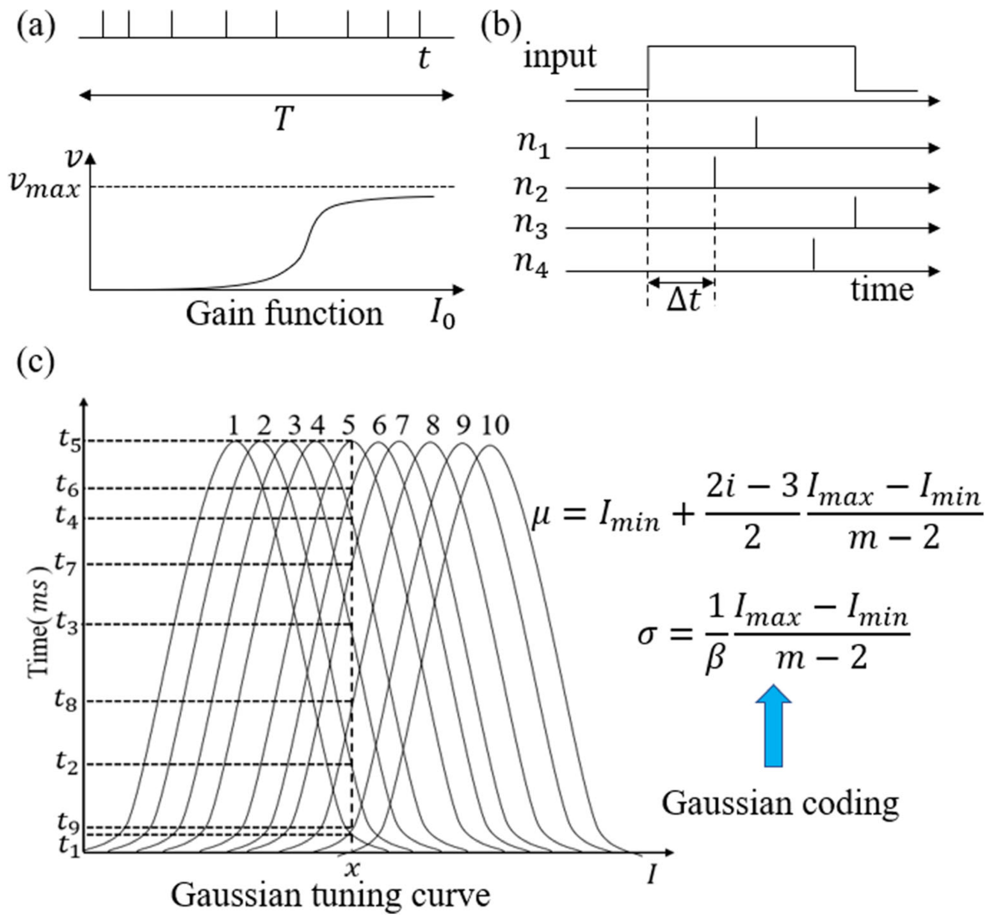


Fig. 5 **a** Frequency coding. **b** Spike time coding. **c** Gaussian coding (The pixel value x in $[a, b]$ is encoded with m neurons. μ and σ are the mean and variance of the Gaussian function corresponding to neuron i)

makes the SNN model more biologically explainable. Later, a neuromorphological chip for SNN was also generated. In the twenty-first century, the emergence of large-scale computing platforms provides a carrier for the simulation and application of SNN. This can play a strong role in promoting the development of SNN.

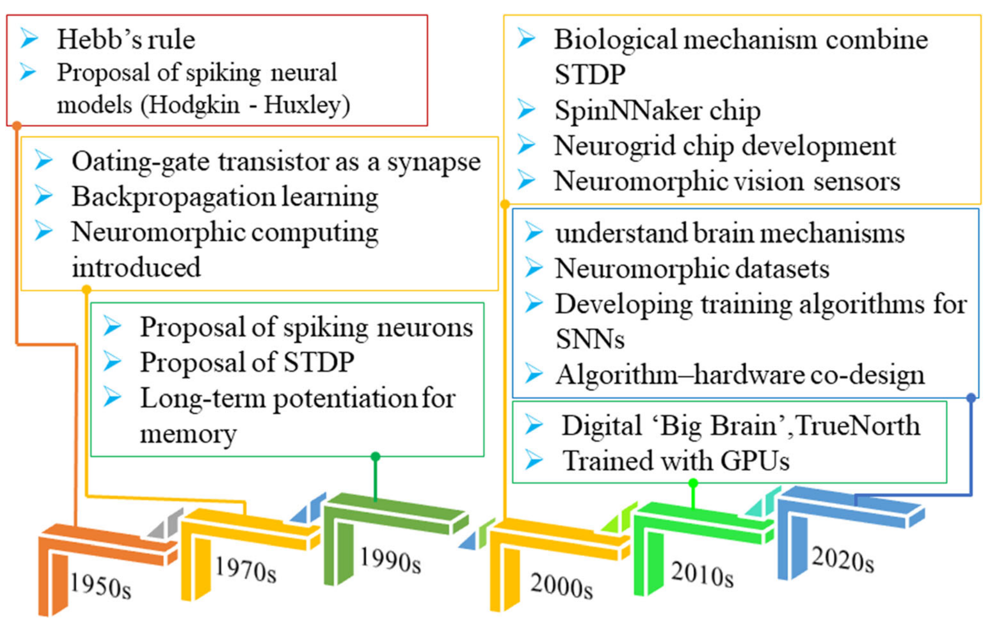


Fig. 6 Development history of SNN

The main differences between SNN and convolutional neural network (CNN) are as follows:

1) The transmission of CNN information depends on high-precision floating-point numbers. Inspired by biological systems, SNN processes spike train information. SNN is an event-driven model, and its basic unit is the spiking neuron.
2) The training and learning of CNNs mainly rely on the BP algorithm based on gradient descent. The learning and memory mechanisms of SNN primarily depend on the synaptic plasticity of presynaptic and postsynaptic neurons.

SNN and CNN are essentially the simulations of biological systems, but their abstractions of biological systems are far from each other. The neural model of CNN is weighted summation followed by nonlinear activation, which is different from the biological brain using spike to transmit information. SNN was born in a background where the biological rationality of CNN was questioned. SNN is more biologically authentic than CNN. At present, SNN learning is very open, and there is no general training algorithm. The design and training of SNN model mainly follow the CNN.

There are direct and indirect training methods for SNN learning. Direct training includes a BP algorithm approximating ANN and a method based on STDP. Indirect training is mainly to convert the trained ANN into SNN through a series of operations. This review mainly selects the research work of SNN in the IP field in the last five years. In addition, we also select some representative research and review articles in the development of SNN. The reference papers in this paper are all closely related to the theme and come from authoritative journals and conferences.

3.1 Direct training

The direct training algorithms of SNN are mainly surrogate gradient learning and STDP. Next, we introduce these two algorithms and review their development in detail.

3.1.1 Surrogate gradient learning

For SNN, the information of the model is expressed in the form of the spike train. The state variables and their error functions inside neurons no longer meet the requirements of continuous differentiability. Using the error BP algorithm of ANN as a reference to construct a surrogate gradient learning algorithm is a way to solve the difficulty of SNN training. The basic framework of this learning algorithm is shown in Fig. 7. It can be seen from the figure that the learning process of SNN can be divided into three stages: first, encode the sample data into a spike train through a specific coding method; second, the spike train $S_i^n(t)$ is input into the model, and a certain simulation strategy is applied to run the neural network to obtain the actual output spike train. Then the error function E between the target output spike train $S_d^m(t)$ and the actual output spike train $S_o^m(t)$ of the neuron is defined, and then the weight is updated using the delta rule $W \leftarrow W + \Delta W$. The definition of SNN error function E is shown in Eq. (6).

$$E = \frac{1}{2}\sum_{m=1}^{N_O}\left(t_o^m - t_d^m\right)^2, \tag{6}$$

where t_o^m and t_d^m represent the actual spike firing time and the target spike firing time of the output layer neuron m respectively. The calculation of the weight is shown in Eq. (7).

$$W = W + \Delta W = W - \eta\nabla E = W - \eta\frac{\partial E}{\partial W}, \tag{7}$$

where η represents the learning rate. ∇E represents the gradient calculation value of error function E to weight W.

Next, we classify and introduce the results of this algorithm.

(1). Improvement based on input coding

Researchers have proposed a series of solutions to the training of SNN around the problem of the non-differentiability of the discrete spike train. These schemes essentially aim at to constructing differentiable SNNs. In 2016, P. O'Connor et al. proposed a deterministic sampling method to encode the input vectors of SNN. The BP approximately solves the gradient of the ReLU to update the weight [57]. Wu et al. proposed an SNN learning rule based on rate-coded, in which the spike count of neurons was used as an alternative to gradient BP [32]. SNN based on rate-coded has achieved success in both multi-layer perceptron (MLP) and CNN. The performance of MLP based on Wu et al. [57] on MNIST is better than that of the fractional-stochastic-gradient-descent (FSGD) algorithm of P. O'Connor [32]. This is because when the model increases, the algorithm based on rate-coded has better applicability than the FSGD.

In 2017, Liu et al. proposed a multi-layer spike neural morphology model based on time and a heuristic loss function. Then they combined it with the derived time error BP. The accuracy of the model on MNIST reached 99.1% [58]. The number of neurons in the model is much lower than that in others [32, 57, 59], but the accuracy of these algorithms on the benchmark dataset is not significant different. Fang et al. proposed a general algorithm based on rate and spatiotemporal mode to train SNN. The performance of this algorithm on neural morphological MNIST, DVS128, TIDIGITS, and other datasets is better than the most advanced methods. In

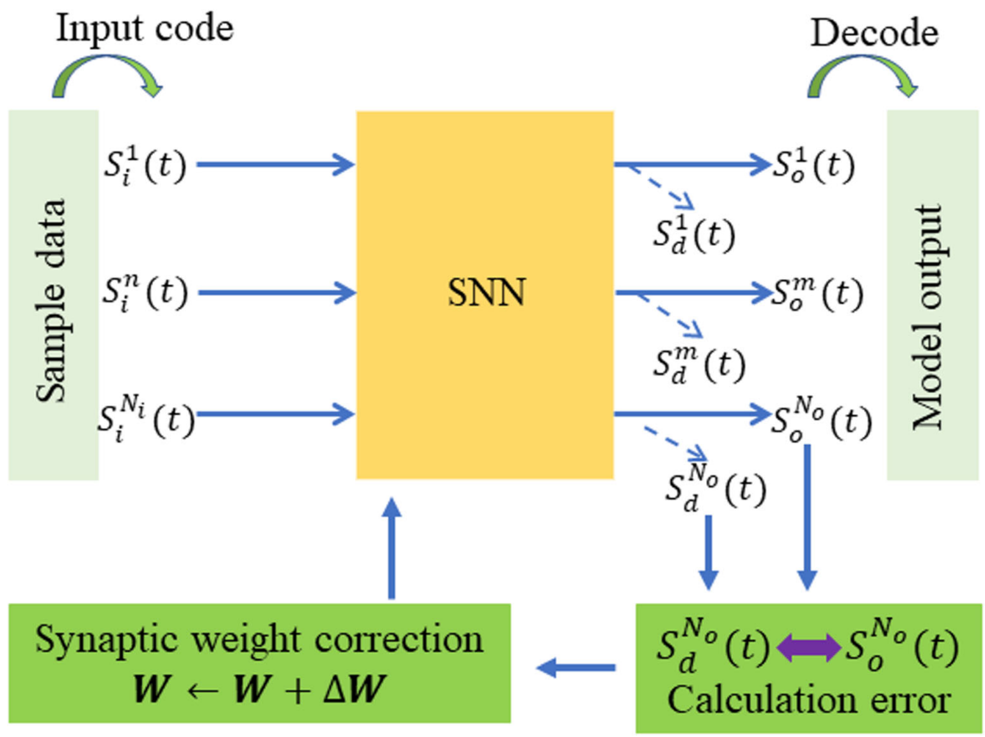


Fig. 7 The basic framework of surrogate gradient learning algorithm

addition to learning and modifying synaptic weights, the algorithm optimizes the spike response core of the synaptic filter to improve the convergence of the model [59]. Although their general algorithm improves the convergence of the model, the time step of input coding is still greater than Wu. [32]. In 2021, Zhou et al. proved that constructing deep SNNs with neurons encoded by a single spike can be trained directly on benchmark datasets. Based on this, they developed a phase-signal processing circuit to implement the neuron, improving energy efficiency by 45% with excellent robustness to time jitter, weight quantization, and noise [60]. The hardware circuit verifies the more energy-saving characteristics of SNN. It can be seen that the improved coding method and neuron model have realized the BP of SNN, and the performance on MNIST can reach the level of ANN.

(2). Constructing approximately differentiable functions

There are also many highlights in improving BP by fitting activation with approximately differentiable functions. For example, Wu et al. (2017) deduced the gradient formula of spatial dimension and time dimension using the mean square error function [61]. In 2019, the above authors first explicitly transformed LIF to obtain its iterative form. Then on this basis, they used the spatio-temporal BP (STBP) training method and rectangular function to approximate differentiation and achieved an accuracy of 99.53% on NMNIST [62]. Similarly, in 2019, Lee et al. removed the LIF leakage term and approximated the activation of neurons as a linear function. Then, they added the estimated compensation gradient of the leakage term to obtain the weight update formula for network training [63]. Compared with others [61–63], the lateral effect between adjacent neurons was introduced into the renewal of membrane potential by Cheng et al. [64]. This lateral action [64] and the method of defining the error solution gradient to update the local weights [65] improves the performance of the model. Both methods can be well expanded in terms of network depth and scale.

An accurate BP algorithm is not crucial for SNN training, which is why the gradient can be approximated by discrete spike information [66–68]. Pehlevan [69] combined non-negative similarity matching cost function to theoretically deduce the unsupervised SNN algorithm. In this algorithm, both sparse feature extraction and manifold learning are expressed as non-negative similarity matching problems. These reasonable approximate equivalent methods benefit the training [70–74]. In 2020, Kim et al. studied two different cross-time domain gradient calculation strategies for SNN training: the gradient method based on activation value and the gradient method based on timing change. Based on comparing the two methods, a new algorithm is proposed, which makes more effective use of each spike [72]. Zhang and Li [73] proposed the TSSL-BP method to train deep SNN, which broke the dependent error BP mechanism. TSSL-BP can effectively train the model in a very small time window. This algorithm achieves 91.47% classification accuracy on CIFAR-10.

Neuromorphological hardware based on SNN has been widely studied in dynamic information processing [75]. Qiao et al. introduced a hardware-friendly binary SNN (BSNN) to effectively identify spatio-temporal event-based data. The recognition accuracy of the trained BSNN on the dataset DVS-CIFAR10, Dvs128 Style, and N-TIDIGITS18 was 62.1%, 97.57%, and 90.35% respectively. The training method of BSNN provides a reference for real-time information processing oriented to neural morphology hardware [76]. In the future, SNN will have important applications in both military and civilian fields [77, 78]. Cheng et al. proposed an improved population coding method for generalized gradient descent training of SNN. Through the analysis of surface EMG signals, the proposed SNN achieves 97.4% recognition accuracy on the gesture recognition dataset FMA [79]. In radar gesture recognition, Safa proposed a novel radar SNN training strategy, which achieved an accuracy of more than 91% on two different radar datasets. They demonstrated the feasibility and power consumption of SNN-based radar processing in the real world. Radar signal processing using SNN is also a solution in the field of ultra-low power human-computer interaction in recent years [80]. The spike-based approximate back propagation (SABP) algorithm and the brain-based generalized SNN framework achieve the best classification accuracy [81] in the small sample dataset sonar image target classification (SITC), in addition to the best classification accuracy on MNIST and CIFAR-10. To solve the classification problem of hyperspectral images (HIS) in the edge computing environment, Liu et al. [82] constructed SNN (SNNSSEM) using IF and SE module networks. The average classification accuracy of SNNSSEM on the three HSI datasets exceeded 99%. Their research explored the application of SNN in hyperspectral remote sensing technology and realized the real-time classification of HSI in the mobile computing environment. Xie et al. [83] proposed an efficient low-energy ship recognition strategy to solve the problems of multiple model parameters and large energy consumption of traditional depth learning in SAR image target recognition.

(3). Surrogate gradient learning combined with STDP

In the BP algorithm, network training combined with STDP is also a popular topic among researchers [84, 85]. In 2018, Lee et al. used STDP for hierarchical unsupervised pre-training of SNN and then used the approximate differential method for follow-up training. The network has an accuracy of 99.28% on MNIST [86]. A. Tavanaei and A. Maida's [87] training algorithm is opposite to the process of Lee [86]. Their algorithm first uses BP to update the weight of SNN and then applies STDP local learning rules to each time step. The model's depth and the network's accuracy on MNIST are not as good as Lee [86]. In addition, there is a study to improve the (back-propagation through time) BPTT of recurrent neural networks (RNNs). For example, Bellec et al. improved the biological irrationality of BPTT in RNN and proposed the E-PROP method to be applied to SNN [88]. The residual structure of the BN layer and the Resnet of the ANN were also improved. In 2021, Zheng et al. proposed a threshold-related batch normalization method to solve the SNN gradient explosion and disappearance problem. The residual structure and emissivity of the network are adjusted based on the STBP-BN method [89]. This method makes it possible to realize deeper SNN training.

The surrogate gradient learning algorithm of SNN is a relatively new research field, and scholars throughout the world have also done abundant research on it. Table 2 compares the performance of BP based SNN algorithm on different datasets. We can see from the table that the application of SNN based on the BP algorithm in the field of image processing mainly focuses on simple classification tasks. Moreover, the network structure of SNN is relatively simple, and its network depth generally does not exceed 9–11 layers [90]. Currently, these algorithms use offline learning, which is only applicable to static data [32, 67]. The spatiotemporal data obtained in the real world shows spatial characteristics at a fixed time, and at the same time, the data generally shows the characteristics of the time train. Therefore, the learning and recognition of spatiotemporal data need to study the online learning algorithm of SNN.

SNN is a kind of computing model similar to the biological nervous system [86]. The backpropagation mechanism combined with STDP is used in image tasks with good performance and applicability [87]. However, this kind of algorithm mainly applies to learn a single neuron or single-layer neural network. Converting discrete spike train into continuous function and interpreting it as the specific physiological signal is the key to realizing the BP algorithm of SNN. The disadvantage of the error function constructed by spike train is that it is only suitable for shallow networks. Due to the inherent complexity of SNN, there are still many difficulties in constructing widely applicable learning algorithms. Furthermore, the new mechanism and algorithm need to be further explored.

3.1.2 STDP mechanism

The nervous system has no external supervisor, so the nervous system, which consists of biological neurons, is

Table 2 Direct training (performance comparison of BP algorithm with approximate gradient)

Network	Method	Dataset	Acc. (%)	Ref.
FCN (784–500–500-10)	Spiking-FSGD	MNIST	97.97	[57]
FCN (784–800-10)	BP with rate-coded	MNIST	98.64	[32]
FCN (784–800–800-10)	BP with rate-coded	MNIST	98.66	[32]
CNN (28×28-12c5-2a-64c5-2a-10)	BP with rate-coded	MNIST	99.26	[32]
FCN (169–500-10)	MT-10 (heu/noheu)	MNIST	99.10	[58]
CNN (32C3-32C3-64C3-P2-64C3-P2–512-10)	IIR-SNN	MNIST	99.46	[59]
CNN (5,32,2-5,16,2–10)	SNN+DT	MNIST	99.33	[60]
FCN (784–800-10)	STBP-SNN	MNIST	98.89	[61]
LeNet	Spike-based BP	MNIST	99.59	[63]
CNN (32C3-P2-32C3-P2–128)	LISNN	MNIST	99.50	[64]
CNN (28×28-12c5-2a-64c5-2a-10)	SpikeCNN	MNIST	99.05	[65]
FCN (784–800-10)	SLTCSNN	MNIST	97.20	[90]
FCN (784–800-10)	Spike-MLP(HM2-BP)	MNIST	98.84	[67]
CNN (15C5-P2-40C5-P2–300-10)	Spike-CNN(HM2-BP)	MNIST	99.42	[67]
CNN (15C5-P2-40C5-P2–300)	ST-RSBP	MNIST	99.57	[71]
FCN (784–800-10)	ANTLR	MNIST	97.63	[72]
CNN (15C5-P2-40C5-P2–300)	TSLL-BP	MNIST	99.53	[73]
CNN (20C2-P2–50C2-P2–200)	Pre-STDP-BP	MNIST	99.28	[86]
FCN (784–500–150-10)	BP-STDP	MNIST	97.20	[87]
CNN (32C3-32C3-64C3-P2-64C3-P2–512-10)	IIR-SNN	NMNIST	99.39	[59]
FCN (784–400–400-10)	STBP-SNN	NMNIST	98.75	[61]
LeNet	Spike-based BP	NMNIST	99.09	[63]
CNN (32C3-P2-32C3-P2–128)	LISNN	NMNIST	99.45	[64]
FCN (2x34x34–800-10)	ANTLR	NMNIST	96.02	[72]
CNN (12C5-P2-64C5-P2)	TSLL-BP	NMNIST	99.35	[73]
VGG-16	SNN+DT	CIFAR-10	92.68	[60]
VGG-9	Spike-based BP	CIFAR-10	90.45	[63]
ResNet-9	Spike-based BP	CIFAR-10	90.35	[63]
ResNet-11	Spike-based BP	CIFAR-10	90.95	[63]
CNN (32x32x3-32c5-2a-32c5-2a-64c4–10)	Spike-CNN	CIFAR-10	75.42	[65]
CNN (128C3-256C3-P2–512C3-P2-1024C3-512C3–1024-512)	TSLL-BP	CIFAR-10	91.47	[73]
GoogleNet	SNN+DT	ImageNet	68.8	[60]

plastic. That is to say, the nervous system uses different types of unsupervised learning. STDP is an SNN weight correction method with biological interpretability. The SNN composed of spiking neurons uses the unsupervised learning rule STDP (described in detail in Chapter 2) to complete the network training. The SNN architecture is divided into two parts: feature extraction and recognition, and classification. The SNN model based on STDP is shown in Fig. 8 (See the **Supplementary materials** for the detailed introduction). The feature extraction part corresponds to each stage of the visual path and extracts the visual features of the input image layer by layer in the process of transmitting the spike signal. In the classification part, first, the spike activity of P_m layer is decoded into feature vectors, and then the SVM classifier is trained to complete the prediction of sample categories. The convolution layer integrates the output information of the previous layer through spike-based convolution operation and uses STDP to extract more abstract and complex visual features. The pooling layer realizes the position invariance of feature extraction through spike-based maximum pooling, removes a large amount of redundant information and improves the transmission efficiency of SNN.

The unsupervised learning STDP mechanism of SNN is theoretically and practically worthy of in-depth study. Due to the limitations of computing resources and platforms, the results of this algorithm are still limited to the simple model.

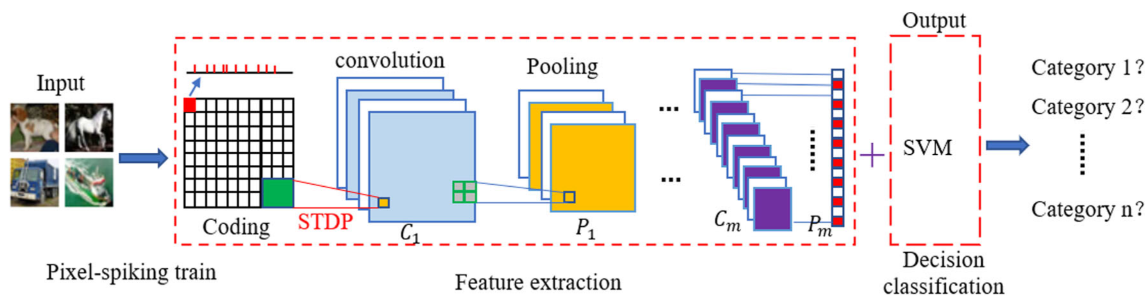


Fig. 8 SNN model based on STDP

Next, we sort out the current work in this field into two categories.

(1). Spike time-dependent plasticity (STDP)

The synaptic strength of biological neurons is related to their spike activity. The presynaptic and postsynaptic spike time is the basis of synaptic strength adjustment. STDP is an abstraction and approximation of this biological mechanism. In 2000, Song et al. proposed the original STDP according to the Hebb rule and gave the corresponding functional representation [18]. In a follow-up study, Masquelier et al. proposed a simplified STDP. The weight updating of this simplified rule depends on the current weight, and the training of presynaptic and postsynaptic neurons is more important than the precise time [14]. On this basis, Mesnard et al. proposed an energy-based local comparison model. This learning algorithm realized balanced propagation in the LIF network [91]. In the above work, the network structure was fixed when classifying and recognizing images. Qi et al. proposed an autonomous learning algorithm. They use the original STDP algorithm to optimize the network, and the tempotron supervision algorithm was used for weight update [84]. The structure of the model can be adjusted according to the needs of the task. Experiments show that this method is superior to the traditional algorithm [14, 18, 91]. The joint learning of network structure and weight improves the robustness of SNN, and the network has higher performance and lower computational costs.

STDP learning mechanism has some limitations in SNN network training. The improved STDP and neuron model can break through the limitations of SNN training to a certain extent. Tavanaei et al. improved based on STDP [85] and divided the network into unsupervised and supervised networks. The 98.6% accuracy reported on the MNIST is better than the existing STDP-based multi-layer ANN. The combination of STDP and delay coding can better understand the learning mode of the visual system. Although the weight-dependent STDP is stable, additional mechanisms need to be introduced, which increases the complexity of SNN [92]. To solve this problem, Shrestha et al. proposed an approximate Bayesian Stochastic-Integrate-and-Fire (SIF) neuron. The network uses STDP and introduces a supervised learning algorithm of teacher-neuron to achieve good accuracy on MNIST [93]. This teacher-neuron improves the network performance without introducing a new learning mechanism. In 2016, Tavanaei et al. [94] slightly improved the probability STDP of Tavanaei [95] and the SNN of Nessler [96] and designed a new layer-by-layer training spike CNN. The network is trained with probabilistic STDP and thresholding LIF model, and the recognition performance is more than 98% on MNIST. After randomly adding Gaussian, salt and pepper noise, the performance loss of SNN is also very small. The introduction of probabilistic STDP breaks the limitation of SNN training layers to a certain extent and increases the robustness. This probabilistic STDP algorithm can make the spike information of the network sparser.

(2). STDP combined with biological mechanism

The original STDP needs to sum up all spike information, and it is very inefficient to use it directly [97, 98]. Therefore, Morrison et al. introduced traces based on STDP and proposed online STDP [99]. Diehl et al. designed a fully connected SNN for handwritten numeral recognition based on online STDP [100], with an accuracy of 95% on MNIST. In 2017, Iyer et al. improved the above author's algorithm to adapt to the new dataset. They proved that replacing MNIST with NMNIST and properly optimizing network parameters can make SNN obtain higher accuracy with fewer neurons [101]. Online STDP can provide a reference for reducing the model and accelerating the training needs. Especially in reducing the model, the advantage of online STDP is obvious.

STDP updates the synaptic weight of neurons based on presynaptic and postsynaptic spikes. If the influence of spike frequency is ignored, this may lead to an overlapping representation of the features learned by the model [102]. Srinivasan et al. proposed an enhanced learning scheme based on membrane potential and spike counting to solve the above problems, improving synaptic learning efficiency [103]. Current visual feedback methods can use spatiotemporal information in the recognition process. However, they cannot handle significant changes. To solve this problem, Liu et al. proposed a deep SNN, which takes the dynamic facial motion as the input of face recognition based on video camouflage [104]. SNN based on online STDP and adaptive threshold achieves 95% classification accuracy in the face database. Saunders et al. [105] and Hazan et al. [106] have done a series of works using the self-organizing mapping (SOM) attribute of the brain and other biologically inspired methods and achieved excellent results. Legenstein et al. proposed the reward mechanism STDP (RM-STDP) based on dopamine [107]. Both SOM and RM-STDP methods are abstracted based on biological mechanisms. It can be seen that the study of biological mechanisms is significant for the development of brain-like intelligence [108].

Some features in the image are essential for recognition, but it is challenging to extract features using STDP. RM-STDP [107] provides a reference for solving the above problems. Next, the researchers carried out relevant research in combination with RM-STDP. For example, Liu et al. solved this problem with RM-STDP [109]. The proposed self-organizing SNN (SOSNN) uses STDP and anti-STDP for network training. Experiments show that SOSNN has advantages in the target recognition, and it also proves that reinforcement learning is an effective way to improve the neural network's performance. Mozafari et al. trained the decision-making layer of SNN based on RM-STDP [110], and the weight update rules used the K-winner-take-all mechanism. This method achieves 97.2% accuracy on MNIST without an external classifier. The computation of SNN is much less than ANN, but it is sensitive to parameters. By adaptively reducing the influence of parameters, SNN may be easier to apply in practice [112, 113]. Falez et al. proposed a new threshold adaptive system [111]. The advantage of this mechanism is that the optimal target value can be set independently for each feature. The proposed model achieves the best results of unsupervised ANN on MNIST (98.60%) and face/motorcycle (99.46%). Currently, SNN is applied to simple image classification tasks, and some algorithms try to use STDP for other tasks [114–116]. For example, Benssassi et al. trained SNN with an unsupervised STDP and applied it to cross-modal tasks [117]. Zhu et al. designed a visual image reconstruction framework similar to the retina. It can flexibly reconstruct the complete texture of the natural scene from spike data [118].

Table 3 compares the performance of the SNN algorithm based on STDP on different data sets. As seen from the table, the network structure of this algorithm is limited to shallow networks, and the depth is generally no more than 4–5 layers. Its main applications focus on the simplest MNIST, and its network performance is far lower than ANN. The advantage of STDP is that it is unsupervised learning and does not need a large amount of label data. As a local weight correction method, STDP has much more computation in SNN than the same type of ANN. The limitation of computing hardware and its demand for computing resources limit the implementation of this algorithm in deep SNN. Unsupervised learning is indispensable in SNN theory. With the popularization of deep learning technology, unsupervised learning will further play its advantages. Building a general and efficient unsupervised learning STDP algorithm requires researchers to focus on breakthroughs.

3.2 Indirect training- ANN to SNN

In recent years, the application of SNN in complex tasks puts forward higher requirements for the depth and performance of the network. At present, ANN-SNN is the most effective scheme to achieve deep SNN. The learning and training of deep SNN usually adopt the transformation from ANN to SNN (ANN-SNN), the conversion diagram is shown in Fig. 9. However, the existing technology cannot convert ANN to SNN equivalently, and the network will cause performance loss in the conversion process [119]. Aiming at the performance loss of ANN-SNN, researchers have proposed a series of threshold balancing algorithms to reduce the loss. This section combs the results of these threshold balancing algorithms, including network improvements, weight normalization methods, and reset mechanisms.

(1). Threshold balancing - network and weight operations

At present, most of the work is focused on reducing the performance loss of the network in the conversion process. In 2013, P. O'Connor et al. used CD offline learning to map the trained network parameters to SNN. Moreover, they gave proof of the concept of migrating offline learning deep belief network (DBN) to SNN for the first time [120]. Their work laid the groundwork for the follow-up study of ANN-SNN. In the same year, J. Perez Carrasco and others used a new method to map the ANN to the event-triggered convolution network, which showed good performance in scene information processing and recognition. However, compared with the non-SNN with the same structure, its classification accuracy has a certain loss [121]. Although similar STDP trains some SNN networks based on unsupervised and self-organizing, this training is still in the initial stage of research [122]. The method of ANN to SNN conversion can more effectively complete the training of deep SNN [123]. This

Table 3 Direct training (Performance of SNN based on STDP on different datasets)

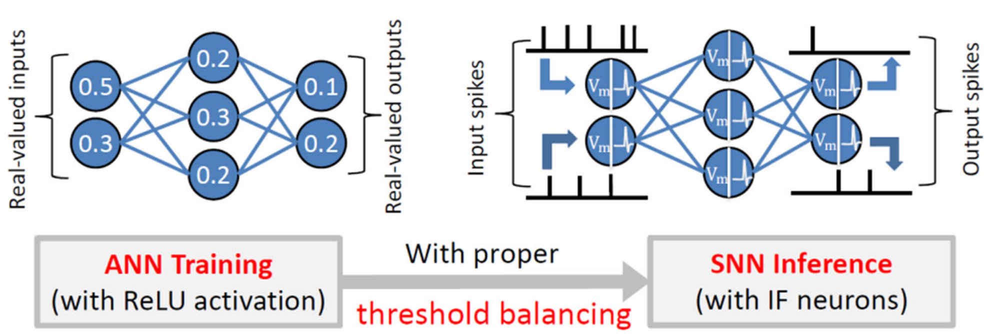
Network	Learning-rule	Dataset	Acc. (%)	Ref.
ConvNets(2Cov.+3FCN)	STDP+BP	MNIST	98.60	[85]
MLP (784–100-10)	SIF-STDP (Q2PS)	MNIST	85.00	[93]
MLP (784–400-10)	SIF-STDP (Q2PS)	MNIST	87.40	[93]
MLP (784–1600-10)	SIF-STDP (Q2PS)	MNIST	89.70	[93]
Spiking CNN(H128-D32)	Probalistic-STDP	MNIST	98.36	[96]
2-Layer (784–6400)	Exponential STDP	MNIST	95.00	[100]
SOSNN	STDP	MNIST	96.10	[109]
SOSNN	RM-STDP	MNIST	97.40	[109]
DCSNN	R-STDP	MNIST	97.20	[110]
2-Layer (784–100)	KSOM Kohonen	MNIST	87.36	[109]
2-Layer (784–100)	DSOM Kohonen	MNIST	85.19	[109]
2-Layer (784–100)	PSOM Kohonen	MNIST	84.53	[109]
FCN (784–4500–10)	VPSNN STDP	MNIST	98.52	[110]
CSNN+SVM	THRADA-STDP	MNIST	98.60	[111]
2-Layer (784-R800)	DVS-STDP	N-MNIST	80.63	[101]

method's biggest challenge is realizing the lossless conversion of network performance.

In 2015, Cao et al. successfully converted a deep SNN with two orders of magnitude lower power consumption than the original ANN by cutting [124]. The converted SNN shows the same performance as ANN on CIFAR-10. Based on the detailed analysis of the above network losses, Diehl et al. discussed the impact of neuron threshold and spike coding frequency on the network and proposed an optimization algorithm to reduce the performance loss [125]. Hunsberger and Eliasmith trained deep SNN using a biologically more reliable LIF model, revealing that the spike firing rate of neurons can produce a level equivalent to that of ANN [126]. They analyzed the loss of network performance in terms of spike firing rate for the first time, creating a precedent for the research on adjusting the spike firing rate and reducing the loss. LIF is a spiking neuron model with a more biological explanation, and the leakage term has a certain regulatory effect on the spike firing rate of SNN. In 2016, Daniel et al. used architectures such as CNN and RNN as classifiers for dynamic vision sensors (DVS). They introduced several pretreatment methods of DVS data for the deep network. Spiking-DBN combines the audio peaks of DVS and dynamic audio sensor (DAS), and currently produces the most advanced performance in visual tasks [127]. Their work has played a role in promoting follow-up research on the combination of chips and SNN.

In ANN, all the neurons must be updated once to get the output. The computational complexity in SNN depends on the number of spikes in the network. A higher spike firing rate and longer computing time usually improve the classification performance of SNN, but higher classification performance can appear earlier. Therefore, in another paper, the author studied how to obtain good network performance with as fast speed and little computation as possible. By minimizing the number of spikes and enhancing the learning of feature detectors, they optimized the deep SNN and realized the rapid classification of the model. The most effective SNN only needs less than 42% of the ANN calculation steps to achieve the target accuracy, while the fastest SNN only needs 25%

Fig. 9 Illustration of the ANN-SNN conversion methodology. Reprinted with permission from ref. [135]



of the input spikes to achieve the same classification accuracy [128]. Their work optimizes the SNN, significantly reduces the delay and computing requirements of the deep network and makes it attractive for applications such as robots. In 2017, based on the conversion method of Cao [124], Li et al. eliminated the truncation error and accumulation error of network conversion by introducing noise during network training. They introduced a bionic spiking neuron to reduce performance loss. SNN achieved 86.43% accuracy on CIFAR-10 [129].

(2). Threshold balancing - improving the neuron model

The work similar to the biological mechanism in the design of neuronal reset helps understand the spike activity of the brain. Introducing a more biological neuron model is key to balancing the firing rate of neurons.

The previous conversion work did not focus on the pooling layer of the network, softmax activation, bias term, and batch processing. The work of Rueckauer et al. fill the gap of previous work. An approximate mathematical theory is given to solve the above problems. They also made simple modifications to the network reset mechanism. The converted network showed good performance on MNIST, CIFAR-10, and ImageNet [130]. Deep SNN is no longer limited to simple classification tasks and is gradually applied to complex tasks. In 2018, Chen et al. successfully converted VGG-19 into SNN using multi-intensity LIF and achieved 94.01% accuracy on CIFAR-10. They also conduct dynamic pruning operations on SNN to construct SM-SNN, which reduces the amount of computation by 85% compared with the original network [131]. Their proposed dynamic pruning mechanism is divided into three stages. The first and second stages removed inactive spike neurons. The third stage severed the weak synapses that contributed little to the output of SNN neurons. Compared with the pruning method [120], this dynamic pruning mechanism effectively reduces the redundant computation in SNN. Moreover, the three-stage dynamic pruning mechanism is more suitable for deep SNN models.

In the above work, a multi-intensity LIF model is used, which relaxes the limitation of spiking neuron output and greatly improves the accuracy of SNN conversion. Zhang et al. also conducted research based on LIF improvement [34]. They proposed the ticking neuron mechanism (as shown in Fig. 10a) according to the characteristics of LIF and used the reverse coding criterion to convert ANN into temporary coding SNN. The network conversion process is shown in Fig. 10b and c. Compared with the original ANN network, the proposed method reduces the computation in SNN by 42%, and the network performance loss is no more than 0.5%. The previous work was to normalize the network's weights on the ANN and then map the weights to the SNN. The threshold balance of the network only performs relevant operations at the ANN level without considering the

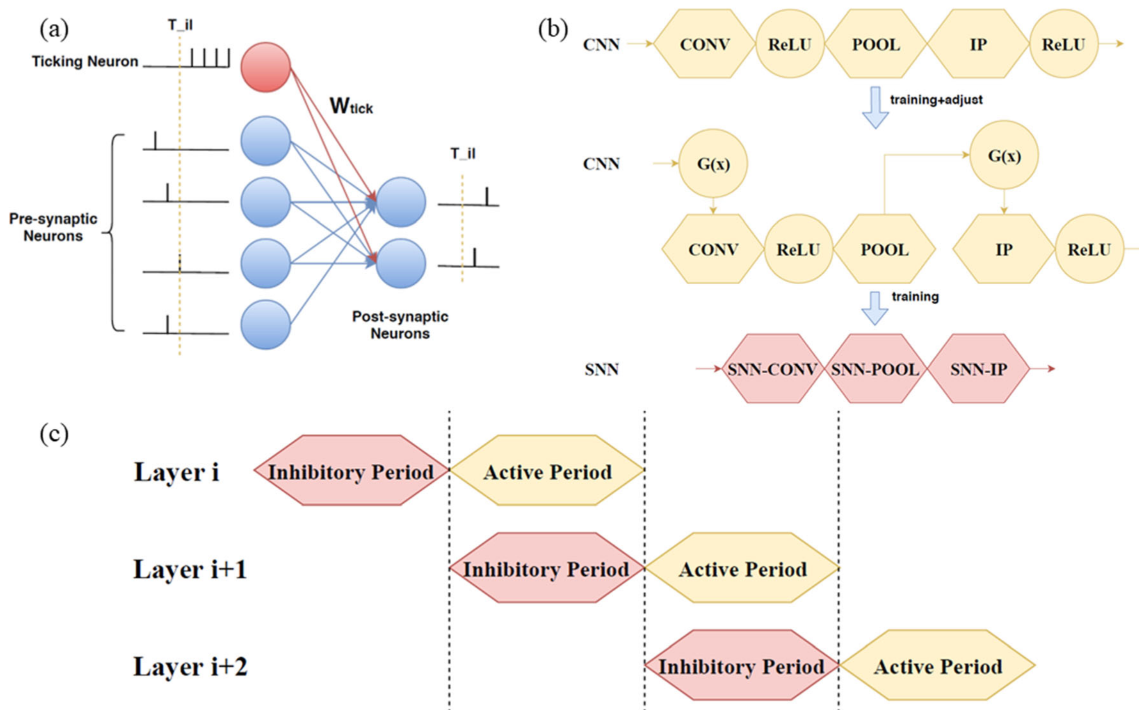


Fig. 10 **a** A Layer with a ticking neuron in SNN. **b** Conversion procedure. **c** Propagation pipeline. Reprinted with permission from ref. [34]

SNN. Sengupta et al. considered the actual operation of SNN in the conversion process and proposed the Spike-NORM weight normalization algorithm. Their Spike-NORM algorithm normalizes the weights according to the maximum spike input received by the neurons in the SNN, which has been applied to VGG-16 and ResNet architectures and achieved good results [132].

Most of the deep SNNs are converted from ANN, and Nitin Rathi et al. took the network weight and threshold obtained by hybrid training as the initialization parameters of SNN. The complexity of network training is reduced, the time steps are reduced, and the convergence speed is accelerated at the same time. VGG-16 obtained by hybrid training achieves 92.02% classification accuracy on CIFAR-10 [133]. In 2020, Han and Roy proposed the temporal-switch coding (TSC) scheme for images and the corresponding TSC spiking neuron model, which converts the deep VGG-16, ResNet-20, and ResNet-34 into the corresponding TSC-SNN, effectively reducing the performance loss [134]. When the membrane potential accumulation of neurons exceeds its threshold, spikes will be emitted, and the membrane potential will be reset.

The previous working membrane potential reset mostly adopts a hard reset, as shown in Fig. 11. The disadvantage of a hard reset is that it will cause a loss of membrane potential. Han et al. showed that the soft reset mechanism of membrane potential (as shown in Fig. 12) effectively reduced the loss of neuronal information. Residual membrane potential (RMP) SNN proposed based on a soft reset mechanism almost realizes lossless conversion [135]. In this paper, VGG-16 is converted to RMP-SNN, the classification accuracy loss is less than 0.01% on CIFAR-10, and the performance loss is only 0.29% and 0.4% on CIFAR-100 and ImageNet. The classification accuracy of RMP-SNN converted by ResNet-20 on CIFAR-100 is 68.72%. Deep SNN has excellent potential in energy saving on special neuromorphological hardware. In the current research, SNN is no longer applied to image classification tasks but is used in the processing of large image data sets and game development [136]. Deng et al. [137] added a threshold to ReLU during

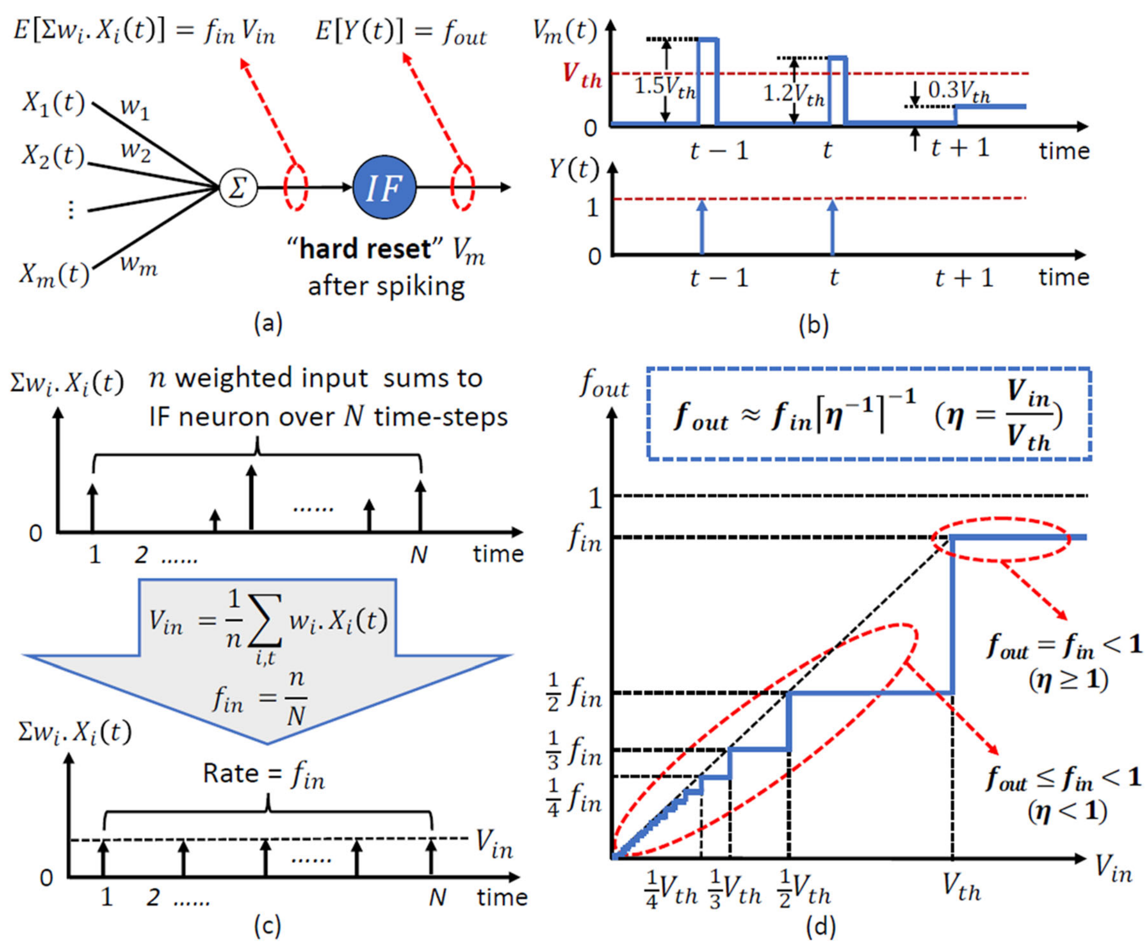


Fig. 11 Hard reset mechanism. Reprinted with permission from ref. [135]

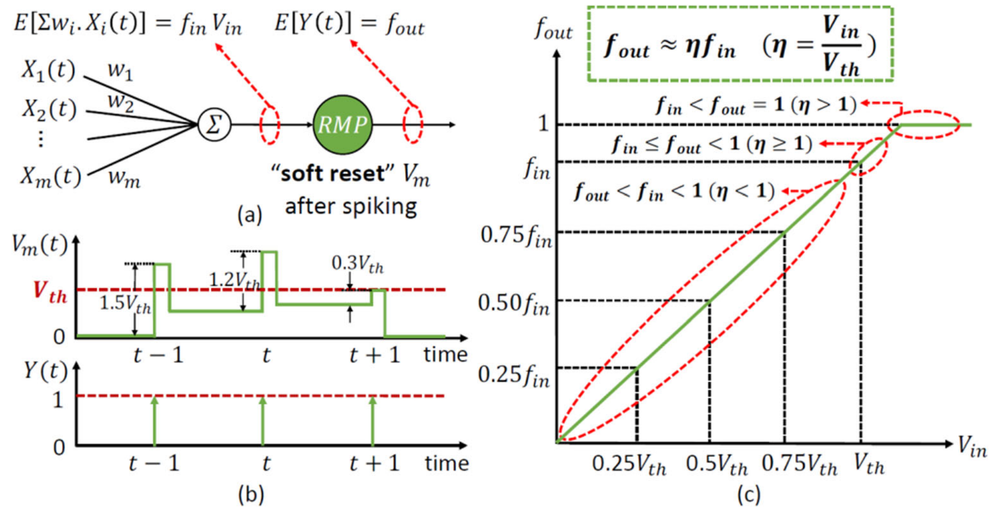


Fig. 12 Soft reset mechanism. Reprinted with permission from ref. [135]

ANN training, and their method effectively reduced the conversion error. Experiments show that this method [137] can achieve good performance of the model with little simulation time.

ANN-SNN algorithm mainly focuses on the threshold balance of the network. Threshold-balance is to reduce the loss of network performance by adjusting the spike firing rate of spiking neurons [138]. The performance of indirectly trained SNNs on different datasets is shown in Table 4. Although the indirect training will bring a certain loss of accuracy to the network, it also solves the problem that SNN is limited to shallow networks [139]. Compared with direct training, ANN-SNN can make the scale of the model larger. The algorithm's performance on CIFAR-10, CIFAR-100, and ImageNet can also reach the same level as ANN [140]. STDP is an unsupervised training algorithm of SNN [141]. Because of its complex dynamics, the STDP algorithm has great limitations in the number of network layers and application scenarios. The alternative gradient learning algorithm has a very low network delay and the encoding time window of input data is relatively small. In model operation, the direct training algorithm pays more attention to the characteristics of the SNN itself, while the ANN-SNN pays more attention to the transformation process. In Table 5, we compare the SNN training algorithms.

4 Other types of SNN

4.1 SNN combined with hardware chip

After decades of development, great progress has also been made in the implementation of SNN on hardware chips [142]. For example, Merolla et al. developed a modular neural synapse core based on parallel and event-driven mode, which realizes the miniaturization of the brain-like neural morphology hardware [143]. Incompatibility between SNN and hardware chip will lead to unsuccessful model construction. To solve the incompatibility between BP and neural morphology chip, K. Esser et al. approximated the SNN discrete spike to a continuous probability value, and then sampled and discretized this continuous probability. They successfully mapped the trained network to the TrueNoth chip and achieved high classification accuracy on MNIST [144]. The work of Merolla et al. [143] and Esser et al. [144] is to train the model in advance and then map the weights to the chip, while Guo et al. [145] proposes a promising online training RK3 method. The RK3 method can achieve accurate, fast and low-energy information processing in real-time applications.

Neural morphology was introduced as a platform for SNN execution. To eliminate the barriers between hardware designers and application developers, Mack et al. [146] proposed a reconfigurable RANC architecture for neural morphological computing. RANC is an open-source and highly flexible ecosystem. *C++* can simulate its experiments in software and FPGA in hardware. Using the RANC framework, the TrueNorth architecture [144] is redesigned and simulated on FPGA. By running SNN to verify the MNIST, the accuracy of the 10-second running time has reached 96.28%. The parallel operation mode of synaptic nuclei in TrueNorth and RANC architecture realizes the rapid processing of information in the design. However, the update of the membrane potential of neurons in the core is calculated in sequence.

Vu et al. [147] proposed a new parallel structure of synaptic nuclei, the neuromorphological structure shown in Fig. 13. The synaptic nucleus of this structure is composed of 256 neurons working at the same time. When working at the same frequency, the architecture of Vu et al. makes the processing speed of MNIST 86 times faster. In contrast, at the same processing speed, the dynamic power of images per second

Table 4 Indirect training (Performance of ANN to SNN algorithm on different datasets)

Network	Conversion Method	Dataset	Acc. (%)	Loss (%)	Ref.
FCN	Model-Based	MNIST	98.61	0.5	[125]
FCN	Data-Based	MNIST	98.64	0.46	[125]
CNN	Soft-LIF	MNIST	98.37	/	[126]
CNN	CNN(LRN/noise)	MNIST	99.09	0.07	[129]
CNN	SNN	MNIST	99.44	<0.01	[128]
VGG-19	M-SNN	MNIST	99.57	/	[131]
Lenet	TDSNN	MNIST	99.08	0.08	[34]
CNN	Visual CNN	N-MNIST	98.30	1.33	[127]
CNN	Soft-LIF	CIFAR-10	82.95	2.42	[126]
CNN	CNN(LRN/noise)	CIFAR-10	86.43	0.05	[129]
VGG-19	M-SNN	CIFAR-10	94.01	/	[131]
VGG-16	SPIKE-NORM	CIFAR-10	91.55	0.15	[132]
ResNet-20	SPIKE-NORM	CIFAR-10	87.46	1.64	[132]
VGG-16	Hybrid Training	CIFAR-10	92.02	/	[133]
ResNet-20	RMP (soft-reset)	CIFAR-10	91.36	0.11	[135]
VGG-16	RMP (soft-reset)	CIFAR-10	93.63	<0.01	[135]
ResNet-20	TSC (time-based)	CIFAR-10	91.42	0.05	[136]
VGG-16	TSC (time-based)	CIFAR-10	93.63	<0.01	[136]
VGG-16	CQ trained SNN	CIFAR-10	92.48	0.08	[142]
VGG-19	CQ trained SNN	CIFAR-10	93.44	0.06	[142]
VGG-*	CQ trained SNN	CIFAR-10	94.16	0.04	[142]
ResNet-20	TSC (time-based)	CIFAR-100	68.18	0.54	[136]
VGG-16	TSC (time-based)	CIFAR-100	70.97	0.25	[136]
ResNet-20	RMP (soft-reset)	CIFAR-100	67.82	0.9	[135]
VGG-16	RMP (soft-reset)	CIFAR-100	70.93	0.29	[135]
VGG-*	CQ trained SNN	CIFAR-100	71.52	0.4	[142]
Alexnet	TDSNN	ImageNet	56.7/79.83	0.46/0.5	[34]
VGG-16	TDSNN	ImageNet	70.87/90.11	1.69/0.83	[34]
VGG-16	SPIKE-NORM	ImageNet	30.04/10.99	0.59/0.38	[132]
ResNet-34	SPIKE-NORM	ImageNet	75.47/86.33	5.22/3.36	[132]
VGG-16	Hybrid Training	ImageNet	65.19	/	[133]
ResNet-20	TSC (time-based)	ImageNet	69.93	0.71	[136]
VGG-16	TSC (time-based)	ImageNet	73.46	0.03	[136]
ResNet-34	RMP (soft-reset)	ImageNet	69.89	0.75	[135]
VGG-16	RMP (soft-reset)	ImageNet	73.09	0.4	[135]

Table 5 Comparison of SNN training algorithms

SNN training	Direct training		Indirect Training
	STDP	Surrogate gradient learning	ANN-SNN
Algorithm	Unsupervised	Supervised	Supervised
Model depth	3–5 layers	9–11 layers	>11 layers
Network Delay	High	Low	High
Application	Image classification (MNIST, NMNIST)	Image classification (MNIST, NMNIST, CIFAR-10)	Image classification (MNIST, NMNIST, CIFAR-10, CIFAR-100, ImageNet), Target tracking, Speech recognition
Operation	Focus on SNN characteristics	Focus on SNN characteristics	Focus on the conversion process

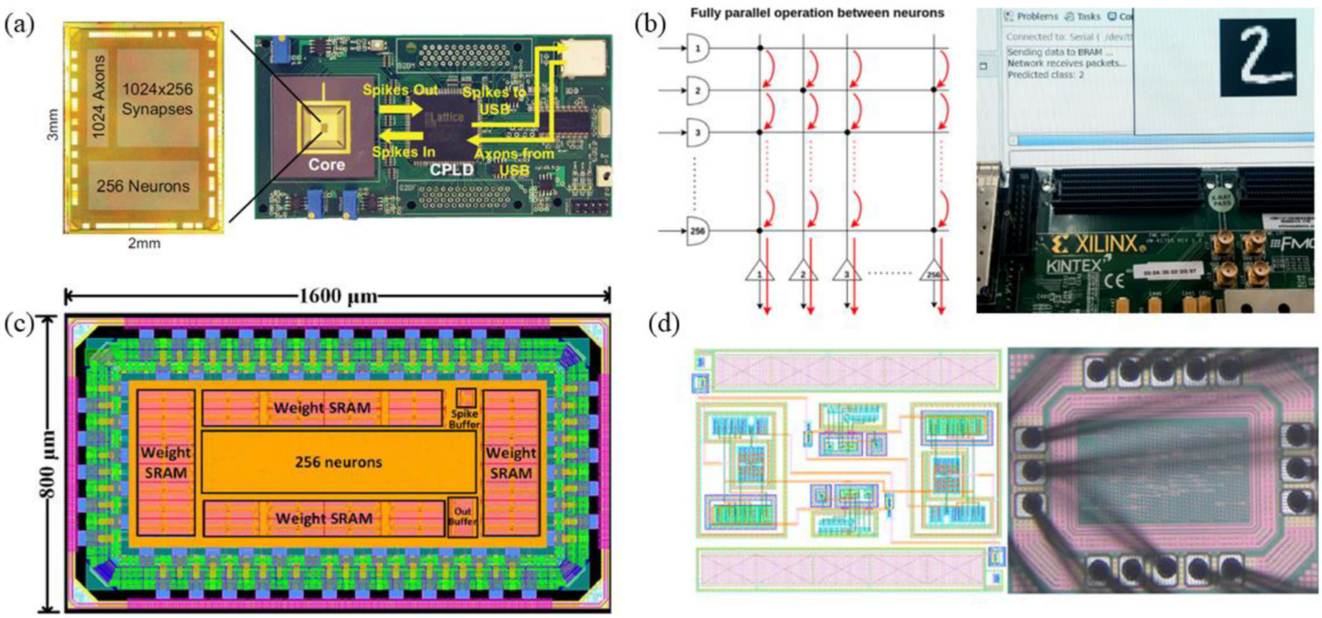


Fig. 13 **a** Neurosynaptic. Reprinted with permission from ref. [144]. **b** Experimental FPGA board running MNIST recognition. Reprinted with permission from ref. [147]. **c** Post-PAR layout of our SNN accelerator. Reprinted with permission from ref. [148]. **d** Layout and chip photo of the LIF and STDP circuits. Reprinted with permission from ref. [149]

is 22 times lower than the result of RANC. The neuron morphology chip is miniaturized, and the smaller chip size integrates a large number of spiking neurons and synapses [148]. The application of SNN in mobile terminal equipment depends on the miniaturization of the spike chip. Famous integrated circuit companies such as Intel, Motorola, and Panasonic have all launched their neural network chips, which have greatly promoted the application of neural networks in the image field. At present, the difficulty of image processing with SNN combined with hardware chip lies in the design of spike chip [149]. Therefore, the design of the chip is the core of the combination of the two.

4.2 SNN combined with new learning mechanism

Since there is no general method and framework for SNN, new algorithms are constantly proposed and applied to practical tasks. To train SNN more effectively and enable it to deal with spatiotemporal spike tasks. A deterministic alternative gradient method super-spike is proposed, which is a nonlinear Hebbian three-factor learning rule [150]. She et al. improved the STDP to correlate the enhancement and suppression of synaptic weight with the frequency of input [151]. The algorithm improves the anti-interference of the network to noise and shows strong resistance to the changes in equipment. Under gradient-based attacks, especially black-box attacks, SNN shows stronger robustness than ANN. Compared with the SNN obtained by ANN conversion, the SNN is directly trained by spike-based BP, which is less affected by the attack [152]. The work of Sharmin shows that small-time windows and LIF neurons play a positive role in improving the robustness of SNN [153].

To enable SNN to solve complex tasks, Kim et al. proposed spiking-Yolo. On Pascal-VOC and MS-COCO, spiking-Yolo achieves the same effect as Tiny-Yolo, and the energy consumption is lower [154]. Chu et al. designed a special SNN processor, which adopted spike-driven processing flow and a hierarchical memory access scheme. They generate a lightweight SNN model through level crossing (LC) sampling, which reduces the spike firing rate of the network and improves efficiency [155]. Mancoo et al. constructed and understood the SNN from the perspective of convex optimization [156]. SNN is proved to generate piecewise linear convex input-output functions. Therefore, SNN can be understood as an arbitrary convex transformation to calculate its input, which is more biologically reasonable than the standard ReLU layer.

The SNN neuromorphological device that simulates brain function is a rapidly developing field [157]. It uses SNN to design a system similar to the human brain to process information efficiently and quickly. Kumarasinghe et al. proposed a brain-inspired SNN (BI-SNN) model [158] in combination with incremental learning. BI-SNN can provide brain-related activities and visual feedback. This study proves the feasibility of discovering muscle activity and limb kinematics from EEG signals. The complex time structure of sensory data and the limitation of computing resources of onboard systems hinder the development of automatic driving algorithms. Lopez Randulfe et al. [159] solved the problem of vehicle-borne radar signal processing by introducing a new SNN to replace the discrete Fourier transform and constant false alarm rate of the original radar data. Compared with the original algorithm, their SNN algorithm has obtained a competitive result in the simulated driving scene. The artificial olfactory system based on SNN and field effect transistor (FET) simulation achieves the detection of toxic gases and the real-time prediction of their concentrations [160]. It can be seen that the security and reliability of the SNN algorithm and its related devices in the actual scenario are particularly important. For the sake of the safety and reliability of the SNN algorithm, the tandem learning rule proposed by Wu et al. reduced the reasoning time and total synaptic operation of SNN by at least one order of magnitude. This cascade learning rule provides a new solution [161] for training highly efficient, low latency, and high-precision deep SNN.

Transformers are widely used in natural language processing because of their high parallelism. At present, Multilayer Perceptron (MLP) combined with transformers is also a popular research topic in the field of visual processing. The key point of utilizing Transformers and MLPs is to divide images into patches and then apply the calculation to each patch. Li et al. [162] proposed a full-precision LIF to realize the communication between patches, including horizontal LIF and vertical LIF in different directions, as shown in Fig. 14a. The authors put this LIF neuron into the MLP and proposed the SNN-MLP, as shown in Fig. 14b. The SNN-MLP achieved a Top1 accuracy of 83.5% on the ImageNet dataset, which was the most advanced result known so far. This work [162] is the first attempt of SNN to combine MLP and transformer and has been successful in the image processing task. In the future, researchers can try to combine LIF neurons with more MLPs and transformers to further improve the performance of new models in detection and segmentation tasks [163, 164].

5 Conclusion

In this review, first, we introduce the development of the neural network, describe the background of SNN, and analyze its information processing mechanism in detail. In the second section of this paper, we review the biological background of SNN, introduce the spiking neuron model, and analyze the differences between different models. We also introduce the STDP rules that simulate biological synaptic behavior and the commonly used coding methods of SNN. In the third section of this paper, we review the research results of SNN in the field of image classification in detail. After review, we conduct an in-depth analysis and detailed comparison of common algorithms of SNN and clarify the advantages and disadvantages of different algorithms. Finally, we have also combed the work of SNN combining hardware chips and new mechanisms. This paper can enable readers to fully understand the learning algorithm of SNN in the field of image classification and the current application results. Our work hopes to provide SNN researchers with more comprehensive research materials and clarify the future research direction of SNN. Hopefully, this review can attract researchers from different disciplines and promote the development of artificial intelligence through interdisciplinary communication and cooperation.

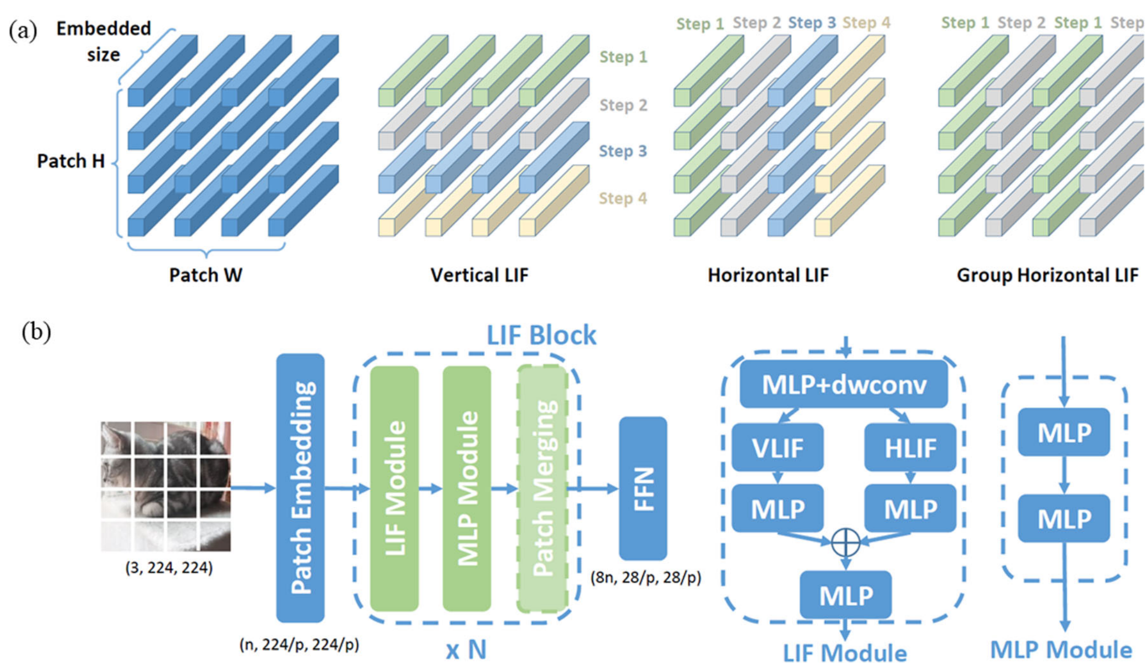


Fig. 14 **a** How apply LIF neurons to the feature maps. **b** Framework of our proposed SNN-MLP. Reprinted with permission from ref. [162]

The SNN algorithm based on STDP and spike-BP focuses on learning shallow networks, so it is challenging to train large-scale deep networks. The method of ANN-SNN solves the technical problem of training deep SNN to a certain extent. ANN-SNN can realize deep and large-scale SNN models, but there is still a lack of effective training algorithms to achieve better accuracy in complex task processing. However, compared with ANN, the performance of SNN in various tasks is still unsatisfactory. With the increase of the converted SNN model, the converted SNN will have an unacceptable approximation error. The SNN model is a new generation of neural networks with more biological interpretability. At present, the proposed technology is not sufficient to expand SNN to deal with complex tasks on a larger scale. Therefore, the realization of deep-seated and large-scale SNN should be a key research direction for researchers, and it is also an urgent problem to be solved. Given the limitations of SNN research, we believe that researchers need to make more remarkable progress and breakthroughs in the following aspects.

1) Biological mechanism.

   (a) Correct understanding of biological actual neural coding is the basis of realizing brain function.
   (b) How to abstract and approximate the spiking neuron model with more biological authenticity and good learning ability according to brain science research is also an urgent problem to be solved to realize large-scale deep-level SNN.
   (c) STDP is the main way of nervous system learning and memory. Building a general and efficient STDP learning mechanism requires researchers to focus on breakthroughs.

2) Design and train SNN.

   SNN supervised learning algorithm is also a new research field. Constructing a widely applicable supervised learning algorithm needs further exploration. The key to research is to design a general modeling and learning framework for SNN based on supervised and unsupervised learning algorithms.

3) Computing platform.

   Because of its advantages of low power consumption, SNN based on hardware chip design can realize a deeper and large-scale network. Therefore, building a neural morphological computing platform with perfect function, high efficiency, and strong universality can promote the development of SNN.

4) Interdisciplinary neuroscience research.

   SNN simulates the parallel information processing mechanism of the brain to varying degrees. Neuroscience researchers have summarized detailed brain models. However, because of its high complexity, it cannot be completely embedded into the real world. Interdisciplinary neural research can greatly promote its rapid development and application in AI.

Neural computing science is based on the research results of the biological neural system and uses computer simulation to complete the processing of complex tasks. The biological nervous system is a multi-level system with a highly complex structure and function. The current level of computer hardware cannot realize the accurate simulation of the nervous system. Based on this situation, researchers should solve this problem in two aspects. (1) A simple neuron calculation model is constructed by analyzing the dynamic characteristics of the neuron spike response. (2) Design a more effective SNN simulation strategy. Due to the high complexity and parallelism of SNN, to quickly and effectively simulate large-scale SNN and meet the needs of solving real-time processing problems, researchers need to study two aspects: parallelization and hardware implementation. (1) The simulation of SNN is realized in a distributed and parallel manner in a multiprocessor system. With the rapid development and broad application of graphics processors, the parallel simulation strategy of SNN is developed on GPU. (2) If SNN is implemented only by software, it is not easy to give full play to the mighty computing power of SNN. Therefore, the hardware implementation of SNN is critical. Researchers can study the simulation process of constructing an SNN on an FPGA chip. According to different spiking neuron models and network topologies, FPGA real-time simulation platforms of various SNNs are realized. Based on FPGA simulation, a special neural computing chip is developed for SNN.

Based on the specific information coding and processing mode of SNN, SNN can be used as a biological neural system modeling tool for simulation and quantitative analysis. SNN will have obvious advantages in the field of Spatio-temporal pattern recognition (speech recognition, target tracking, industrial fault detection, etc.). SNN can become a reliable auxiliary diagnostic method in the medical field. Because of the unpredictability of the human body and disease, biological signals have complex nonlinear relationships in their manifestations and changing laws. SNN is an adaptive dynamic system composed of a large number of neurons, which can solve difficult problems in medical signal analysis. The biological signals output by most medical detection devices also have temporal characteristics, so building an SNN-based biological signal detection and processing model is of great value for clinical diagnosis. This kind of model can be used for ECG, EEG, auditory signal extraction, and medical image processing. SNN can also be used in the control process of a robot or agent. By adding an adaptive learning mechanism, the robot can reflect intelligent behavior and perform better in the interactive environment.

The advantages of SNN are embodied in a system with sensors, chips, and powerful SNN algorithms. The SNN model aims to study adaptive and nonprogrammed brain information processing methods, constantly explore more scientific biological mechanisms and realize brain-like intelligence. Its power consumption and running speed are unmatched by ANN. Therefore, the research on models with more biological authenticity and good learning ability is still the main work of SNN research. The research of biological systems is the foundation for promoting the development of SNN, and the research of brain-like chips is the key to the application of SNN. Therefore, the development of SNN needs multi-faceted coordination.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10489-023-04553-0.

Acknowledgements This work is supported by National Nature Science Foundation of China (grant No.61871106), and the Open Project Program Foundation of the Key Laboratory of Opto-Electronics Information Processing, Chinese Academy of Sciences (OEIP-O-202002).

Declarations

Conflict of interest The authors declare that they have no conflict of interest

References

1. Azevedo FA et al (2009) Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. J Comp Neurol 513(5):532–541
2. Maass W (1997) Networks of spiking neurons: the third generation of neural network models. Neural Netw 10(9):1659–1671
3. Taherkhani A, Belatreche A, Li Y, Cosma G, Maguire LP, McGinnity TM (2020) A review of learning in biologically plausible spiking neural networks. Neural Netw 122:253–272
4. Tan C, Šarlija M, Kasabov N (2020) Spiking neural networks: background, recent development and the NeuCube architecture. Neural Process Lett 52(2):1675–1701
5. Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal of Physiology, vol. 117
6. Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate cortex. The Journal of Physiology, vol. 148
7. Bing Z, Meschede C, Rohrbein F, Huang K, Knoll AC (2018) A survey of robotics control based on learning-inspired spiking neural networks. Front Neurorobot 12:35
8. Panchev C, Wermter S (2004) Spike-timing-dependent synaptic plasticity: from single spikes to spike trains. Neurocomputing 58-60:365–371
9. Wysoski SG, Benuskova L, Kasabov N (Sep 2010) Evolving spiking neural networks for audiovisual information processing. Neural Netw 23(7):819–835
10. Tang C, Chehayeb D, Srivastava K, Nemenman I, Sober SJ (2014) Millisecond-scale motor encoding in a cortical vocal area. PLoS Biol 12(12):e1002018
11. Serre T (2014) Hierarchical Models of the Visual System In: Encyclopedia of Computational Neuroscience. pp. 1–12
12. Tavanaei A, Ghodrati M, Kheradpisheh SR, Masquelier T, Maida A (Mar 2019) Deep learning in spiking neural networks. Neural Netw 111:47–63
13. Tavanaei A, Maida A (2017) Bio-inspired Multi-layer Spiking Neural Network Extracts Discriminative Features from Speech Signals. In: Neural Information Processing (Lecture Notes in Computer Science). pp. 899–908
14. Masquelier T, Thorpe SJ (2005) Unsupervised learning of visual features through Spike Timing Dependent Plasticity. PLoS Computational Biology, vol. preprint, no. 2007
15. Perrinet L, Delorme A, Samuelides M, Thorpe SJ (2001) Networks of integrate-and-fire neuron using rank order coding a: how to implement spike time dependent Hebbian plasticity. Neurocomputing 38-40:817–822
16. Abraham WC, Tate WP (1997) Metaplasticity: a new vista across the field of synaptic plasticity. Prog Neurobiol 52(4):303–323
17. Rossum MV (2000) Stable Hebbian learning from spike timing-dependent plasticity. The Journal of Neuroscience, vol. 20, no. 23
18. Song S, Miller KD, Abbott LF (2000) Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nat Neurosci 3(9):919–926
19. Beyeler M, Dutt ND, Krichmar JL (Dec 2013) Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule. Neural Netw 48:109–124
20. Meftah B, Lezoray O, Benyettou A (2010) Segmentation and edge detection based on spiking neural network model. Neural Process Lett 32(2):131–146
21. Thorpe SJ, Guyonneau R, Guilbaud N, Allegraud J-M, VanRullen R (2004) SpikeNet: real-time visual processing with one spike per neuron. Neurocomputing 58-60:857–864
22. Lee J, Aphinyanaphongs Y, Curtin JP, Chern JY, Frey MK, Boyd LR (Sep 2016) The safety of same-day discharge after laparoscopic hysterectomy for endometrial cancer. Gynecol Oncol 142(3): 508–513
23. Samanwoy GD, Hojjat A (2007) Improved spiking neural networks for EEG classification and epilepsy and seizure detection. Integrat Comput-Aided Engin 14(3):187–212
24. Kasabov N, Feigin V, Hou ZG, Chen Y, Liang L, Krishnamurthi R, Othman M, Parmar P (2014) Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke. Neurocomputing 134:269–279
25. Kasabov NK (Apr 2014) NeuCube: a spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. Neural Netw 52:62–76
26. Pfeiffer M, Pfeil T (2018) Deep learning with spiking neurons: opportunities and challenges. Front Neurosci 12:774
27. Praveena J, Venkatesan K (2014) Advanced auto adaptive edge-detection algorithm for flame monitoring and fire image processing
28. Querlioz D, Bichler O, Dollfus P, Gamrat C (2013) Immunity to device variations in a spiking neural network with Memristive Nanodevices. IEEE Trans Nanotechnol 12(3):288–295
29. Brader JM, Senn W, Fusi S (2007) Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. Neural Comput 19(11):2881–2912
30. Moraitis T, Sebastian A, Eleftheriou E (2018) Spiking neural networks enable two-dimensional neurons and unsupervised multi-timescale learning. In: 2018 International Joint Conference on Neural Networks (IJCNN)
31. Luke R, Mcalpine D (2019) A Spiking Neural Network Approach to Auditory Source Lateralisation. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)
32. Wu J, Chua Y, Zhang M, Yang Q, Li G, Li H (2019) Deep spiking neural network with spike count based learning rule: IEEE

33. Cruz-Albrecht JM, Yung MW, Srinivasa N (Jun 2012) Energy-efficient neuron, synapse and STDP integrated circuits. IEEE Trans Biomed Circuits Syst 6(3):246–256
34. Zhang L, Zhou S, Zhi T, Du Z, Chen Y (2019) TDSNN: from deep neural networks to deep spike neural networks with temporal-coding. Proc AAAI Conf Artificial Intel 33:1319–1326
35. Lee C, Kosta AK, Zhu AZ, Chaney K, Daniilidis K, Roy K (2020) Spike-FlowNet: Event-based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks. Spike-FlowNet: Event-Based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks
36. Amirsoleimani A, Ahmadi M, Ahmadi A (2017) STDP-based unsupervised learning of memristive spiking neural network by Morris-Lecar model. In: 2017 International Joint Conference on Neural Networks (IJCNN)
37. Moraitis T, Sebastian A, Boybat I, Gallo ML, Eleftheriou E (2017) Fatiguing STDP: learning from spike-timing codes in the presence of rate codes. In: 2017 International Joint Conference on Neural Networks (IJCNN)
38. Yu Q, Wang L, Dang J (2018) Efficient Multi-spike Learning with Tempotron-Like LTP and PSD-Like LTD. In: Neural Information Processing (Lecture Notes in Computer Science). pp. 545–554
39. Alemi A, Machens C, Denève S, Slotine JJ (2017) Learning arbitrary dynamics in efficient, balanced spiking networks using local plasticity rules
40. Kang WM, Kim CH, Lee S, Woo SY, Lee JH (2019) A spiking neural network with a global self-controller for unsupervised learning based on spike-timing-dependent plasticity using flash memory synaptic devices. In; 2019 International Joint Conference On Neural Networks (IJCNN)
41. Kiselev M, Lavrentyev A (2019) A preprocessing layer in spiking neural networks – structure, parameters, performance criteria. In: 2019 International Joint Conference on Neural Networks (IJCNN)
42. Hu Z, Wang T, Hu X (2017) An STDP-Based Supervised Learning Algorithm for Spiking Neural Networks. In: Neural Information Processing (Lecture Notes in Computer Science. pp. 92–100
43. Zuo L, Chen S, Qu H, Zhang M (2017) A Fast Precise-Spike and Weight-Comparison Based Learning Approach for Evolving Spiking Neural Networks. In: Neural Information Processing (Lecture Notes in Computer Science. pp. 797–804
44. Matsubara T (2017) Spike timing-dependent conduction delay learning model classifying spatio-temporal spike patterns. In: international joint conference on neural networks
45. Falez P, Tirilly P, Bilasco IM, Devienne P, Boulet P (2018) Mastering the output frequency in spiking neural networks. In: 2018 International Joint Conference on Neural Networks (IJCNN)
46. Chen R, Ma H, Guo P, Xie S, Li P, Wang D (2018) Low Latency Spiking ConvNets with Restricted Output Training and False Spike Inhibition. pp. 1–8
47. Slade S, Zhang L (2018) Topological evolution of spiking neural networks," in 2018 international joint conference on neural networks (IJCNN)
48. Taherkhani A, Cosma G, McGinnity TM (2020) Optimization of output spike train encoding for a spiking neuron based on its Spatio–temporal input pattern. IEEE Transac Cog Develop Syst 12(3):427–438
49. Bagheri A, Rajendran O, Rajendran B (2017) Training probabilistic spiking neural networks with first-to-spike decoding
50. Kumar S, Kaposvari P, Vogels R (Aug 2017) Encoding of predictable and unpredictable stimuli by inferior temporal cortical neurons. J Cogn Neurosci 29(8):1445–1454
51. Thorpe S, Fize D, Marlot C (n.d.) Speed of processing in the human visual system. Nature
52. Nadasdy Z (2009) Information encoding and reconstruction from the phase of action potentials. Front Syst Neurosci 3:6
53. Loiselle S, Rouat J, Pressnitzer D, Thorpe S (2005) Exploration of rank order coding with spiking neural networks for speech recognition. In: IEEE international joint conference on neural networks. pp. 2076-2080
54. Freiwald WA, Tsao DY (2010) Functional compartmentalization and viewpoint generalization within the macaque face-processing system. Science 330(6005):845–851
55. Yepes AJ, Tang J, Mashford BS (2017) Improving Classification Accuracy of Feedforward Neural Networks for Spiking Neuromorphic Chips. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17). pp. 1973–1979
56. Roy K, Jaiswal A, Panda P (2019) Towards spike-based machine intelligence with neuromorphic computing. Nature 575(7784): 607–617
57. O'Connor P, Welling M (2016) Deep Spiking Networks
58. Tao L, Liu Z, Lin F, Jin Y, Gang Q, Wen W (2017) MT-spike: a multilayer time-based spiking neuromorphic architecture with temporal error backpropagation. IEEE Press
59. Fang H, Shrestha A, Zhao Z, Qiu Q (2020) Exploiting neuron and synapse filter dynamics in spatial temporal learning of deep spiking neural network. In Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence {IJCAI-PRICAI-20}
60. Zhou S, Li X, Chen Y, Chandrasekaran ST, Sanyal A (2019) Temporal-coded deep spiking neural network with easy training and robust performance
61. Wu Y, Deng L, Li G, Zhu J, Shi L (2017) Spatio-temporal backpropagation for training high-performance spiking neural networks
62. Wu Y, Deng L, Li G, Zhu J, Shi L (2019) Direct training for spiking neural networks: faster, larger, better. Proc AAAI Conf Artificial Intel 33:1311–1318
63. Lee C, Sarwar SS, Panda P, Srinivasan G, Roy K (2019) Enabling spike-based backpropagation for training deep neural network architectures
64. Cheng X, Hao Y, Xu J, Xu J (2020) LISNN: improving spiking neural networks with lateral interactions for robust object recognition. In: Twenty-Ninth International Joint Conference On Artificial Intelligence and Seventeenth Pacific rim International Conference on Artificial Intelligence {IJCAI-PRICAI-20}
65. Panda P, Roy K (2016) Unsupervised regenerative learning of hierarchical features in spiking deep networks for object recognition. In: 2016 international joint conference on neural networks (IJCNN)
66. Neftci EO, Augustine C, Paul S, Detorakis G (2017) Event-driven random Back-propagation: enabling neuromorphic deep learning machines. Front Neurosci 11:324
67. Jin Y, Zhang W, Li P (2018) Hybrid macro/Micro level backpropagation for training deep spiking neural networks
68. Huh D, Sejnowski TJ (2017) Gradient descent for spiking neural networks
69. Pehlevan C (2019) A spiking neural network with local learning rules derived from nonnegative similarity matching. In: IEEE
70. Zhang M, Wu J, Chua Y, Luo X, Li H (2019) MPD-AL: an efficient membrane potential driven aggregate-label learning algorithm for spiking neurons
71. Zhang W, Li P (2019) Spike-train level backpropagation for training deep recurrent spiking neural networks
72. Kim J, Kim K, Kim JJ (2020) Unifying activation- and timing-based learning rules for spiking neural networks
73. Zhang W, Li P (2020) Temporal spike sequence learning via backpropagation for deep spiking neural networks
74. Wu H, Zhang Y, Weng W, Xiong YZZ, Zha Z-J, Sun X, Wu F (2021) Training Spiking Neural Networks with Accumulated

Spiking Flow", Association for the Advancement of Artificial Intelligence(AAAI)

75. Hu S, Qiao G, Chen T, Yu Q, Liu Y, Rong L (2021) Quantized STDP-based online-learning spiking neural network. Neural Comput & Applic 33(19):12317–12332
76. Qiao GC, Ning N, Zuo Y, Hu SG, Yu Q, Liu Y (Oct 7 2021) Direct training of hardware-friendly weight binarized spiking neural network with surrogate gradient learning towards spatio-temporal event-based dynamic data recognition. Neurocomputing 457:203–213
77. Chen J, Qiu X, Ding C, Wu Y (2022) SAR image classification based on spiking neural network through spike-time dependent plasticity and gradient descent. ISPRS J Photogramm Remote Sens 188:109–124
78. Zhang F, Zhong Y, Chen L, Wang Z (2022) Event-based circular detection for AUV docking based on spiking neural network. Frontiers in Neurorobotics, vol. 15, Art no 815144, 15
79. Cheng L, Liu Y, Hou Z-G, Tan M, Du D, Fei M (2019) A rapid spiking neural network approach with an application on hand gesture recognition. IEEE Transac Cog Develop Syst 13(1):151–161
80. Safa A, Bourdoux A, Ocket I, Catthoor F, Gielen GG (2021) On the use of spiking neural networks for ultralow-power radar gesture recognition. IEEE Microw Wireless Compo Lett 32(3):222–225
81. Liu Y et al (2022) Spike-based approximate backpropagation algorithm of brain-inspired deep SNN for sonar target classification. Computational Intelligence and Neuroscience, vol, 2022
82. Liu Y, Cao K, Wang R, Tian M, Xie Y (2022) Hyperspectral Image Classification of Brain-Inspired Spiking Neural Network Based on Attention Mechanism. IEEE Geoscience and Remote Sensing Letters, vol. 19, 2022, Art no. 6009405
83. Xie H, Jiang X, Hu X, Wu Z, Wang G, Xie K (2022) High-efficiency and low-energy ship recognition strategy based on spiking neural network in SAR images. Frontiers in Neurorobotics, vol. 16. Art no. 970832
84. Yu Q, Shen J, Wang Y, Tang H, Gang P (2018) Jointly learning network connections and link weights in spiking neural networks. In: Twenty-Seventh International Joint Conference on Artificial Intelligence {IJCAI-18}
85. Tavanaei A, Kirby Z, Maida AS (2018) "training spiking ConvNets by STDP and gradient descent," in 2018 international joint conference on neural networks (IJCNN)
86. Lee C, Panda P, Srinivasan G, Roy K (2018) Training deep spiking convolutional neural networks with STDP-based unsupervised pre-training followed by supervised fine-tuning. Front Neurosci 12:435
87. Tavanaei A, Maida A (2019) BP-STDP: approximating backpropagation using spike timing dependent plasticity. Neurocomputing 330:39–47
88. Bellec G, Scherr F, Hajek E, Salaj D, Legenstein R, Maass W (2019) Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets
89. Zheng H, Wu Y, Deng L, Hu Y, Li G (2020) Going deeper with directly-trained larger spiking neural networks
90. Mostafa H (Jul 2018) Supervised learning based on temporal coding in spiking neural networks. IEEE Trans Neural Netw Learn Syst 29(7):3227–3235
91. Mesnard T, Gerstner W, Brea J (2016) Towards deep learning with spiking neurons in energy based models with contrastive Hebbian plasticity
92. Querlioz D, Bichler O, Gamrat C (2018) Simulation of a memristor-based spiking neural network immune to device variations. In: IEEE
93. Shrestha A, Ahmed K, Wang Y, Qiu Q (2017) Stable spike-timing dependent plasticity rule for multilayer unsupervised and supervised learning: In: 2017 International Joint Conference on Neural Networks (IJCNN)
94. Tavanaei A, Masquelier T, Maida AS (2016) Acquisition of Visual Features through Probabilistic Spike-Timing-Dependent Plasticity0
95. Nessler B, Pfeiffer M, Buesing L, Maass W (2013) Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. PLoS Comput Biol 9(4):e1003037
96. Tavanaei A, Maida AS (2017) Multi-layer unsupervised learning in a spiking convolutional neural network. In: International Joint Conference on Neural Networks
97. Thiele JC, Bichler O, Dupret A (2018) A timescale invariant STDP-based spiking deep network for unsupervised online feature extraction from event-based sensor data," in 2018 international joint conference on neural networks (IJCNN)
98. Yu M, Tang H, Gang P (2018) A supervised multi-spike learning algorithm for spiking neural networks. In: 2018 international joint conference on neural networks (IJCNN)
99. Morrison A, Aertsen A, Diesmann M (2007) Spike-timing-dependent plasticity in balanced random networks. Neural Comput 19(6):1437–1467
100. Diehl PU, Cook M (2015) Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Front Comput Neurosci 9:99
101. Iyer LR, Basu A (2017) Unsupervised learning of event-based image recordings using spike-timing-dependent plasticity. In: international joint conference on neural networks
102. Xiao R, Yu Q, Yan R, Tang H (2019) Fast and accurate classification with a multi-spike learning algorithm for spiking neurons," In: Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}
103. Srinivasan G, Roy S, Raghunathan V, Roy K (2017) Spike timing dependent plasticity based enhanced self-learning for efficient pattern recognition in spiking neural networks. In: 2017 International Joint Conference on Neural Networks (IJCNN)
104. Liu D, Yue S (2018) Video-based disguise face recognition based on deep spiking neural network. In: 2018 International Joint Conference on Neural Networks (IJCNN)
105. Saunders DJ, Siegelmann HT, Kozma R, Ruszinkao M (2018) STDP learning of image patches with convolutional spiking neural networks," in 2018 international joint conference on neural networks (IJCNN)
106. Hazan H, Saunders D, Sanghavi DT, Siegelmann H, Kozma H (2018) Unsupervised Learning with Self-Organizing Spiking Neural Networks. pp. 1–6
107. Legenstein R, Pecevski D, Maass W (2008) A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. PLoS Comput Biol 4(10):e1000180
108. Legenstein R, Chase SM, Schwartz AB, Maass W (2010) A reward-modulated hebbian learning rule can explain experimentally observed network reorganization in a brain control task. J Neurosci 30(25):8400–8410
109. Liu J, Zhao G (2018) A bio-inspired SOSNN model for object recognition. pp. 1–8
110. Mozafari M, Ganjtabesh M, Nowzari-Dalini A, Thorpe SJ, Masquelier T (2019) Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks. Pattern Recogn 94:87–95
111. Falez P, Tirilly P, Bilasco IM, Devienne P, Boulet P (2019) Multi-layered spiking neural network with target timestamp threshold adaptation and STDP
112. Khacef L, Miramond B, Barrientos D, Upegui A (2019) Self-organizing neurons: toward brain-inspired unsupervised learning," in IEEE international joint conference on neural networks (IJCNN) 2019

113. Xiang S, Ren Z, Song Z, Zhang Y, Hao Y (2020) Computing Primitive of Fully VCSEL-Based All-Optical Spiking Neural Network for Supervised Learning and Pattern Classification. IEEE Trans Neural Netw Learn Syst PP(99):1–12
114. Wu J, Chua Y, Li H (2018) A biologically plausible speech recognition framework based on spiking neural networks. In: 2018 international joint conference on neural networks (IJCNN)
115. Yang Y, Ren J, Duan F (2022) The spiking rates inspired encoder and decoder for spiking neural networks: an illustration of hand gesture recognition. Cogn Comput:1–16
116. Yao Y, Yu Q, Wang L, Dang J (2019) A spiking neural network with distributed Keypoint encoding for robust sound recognition. In: 2019 international joint conference on neural networks (IJCNN)
117. Mansouri-Benssassi E, Ye J (2019) Speech emotion recognition with early visual cross-modal enhancement using spiking neural networks," in 2019 international joint conference on neural networks (IJCNN)
118. Zhu L, Dong S, Li J, Huang T, Tian Y (2020) Retina-Like Visual Image Reconstruction via Spiking Neural Model. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
119. Maass W, Markram H (2004) On the computational power of circuits of spiking neurons. J Comput Syst Sci 69(4):593–616
120. O'Connor P, Neil D, Liu SC, Delbruck T, Pfeiffer M (2013) Real-time classification and sensor fusion with a spiking deep belief network. Front Neurosci 7:178
121. Perez-Carrasco JA, Bo Zhao, Serrano C, Acha B, Serrano-Gotarredona T, Shouchun Chen, Linares-Barranco B (Nov 2013) Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward ConvNets. IEEE Trans Pattern Anal Mach Intell 35(11):2706–2719
122. Esser SK et al (2016) Convolutional networks for fast, energy-efficient neuromorphic computing. Proc Natl Acad Sci U S A 113(41):11441–11446
123. Hunsberger E, Eliasmith C (2016) Training spiking deep networks for neuromorphic hardware
124. Cao Y, Chen Y, Khosla D (2014) Spiking deep convolutional neural networks for energy-efficient object recognition. Int J Comput Vis 113(1):54–66
125. Diehl PU, Neil D, Binas J, Liu M, Liu SC (2015) Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In: International Joint Conference on Neural Networks
126. Hunsberger E, Eliasmith C (2015) Spiking deep networks with LIF neurons. Computer ence
127. Neil D, Liu S-C (2016) Effective sensor fusion with event-based sensors and deep network architectures," presented at the 2016 IEEE international symposium on circuits and systems (ISCAS)
128. Neil D, Pfeiffer M, Liu S-C (2016) Learning to be efficient. Presented at the proceedings of the 31st annual ACM symposium on applied computing
129. Li J, Hu W, Yuan Y, Huo H, Fang T (2017) Bio-Inspired Deep Spiking Neural Network for Image Classification. In: Neural Information Processing (Lecture Notes in Computer Science. pp. 294–304
130. Rueckauer B, Lungu IA, Hu Y, Pfeiffer M, Liu SC (2017) Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. Front Neurosci 11:682
131. Chen R, Ma H, Xie S, Guo P, Li P, Wang D (2018) Fast and Efficient Deep Sparse Multi-Strength Spiking Neural Networks with Dynamic Pruning. pp. 1–8
132. Sengupta A, Ye Y, Wang R, Liu C, Roy K (2019) Going deeper in spiking neural networks: VGG and residual architectures. Front Neurosci 13:95
133. Rathi N, Srinivasan G, Panda P, Roy K (2020) Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation
134. Han BRoy K (2020) Deep spiking neural network: energy efficiency through time based coding. Computer Vision – ECCV, 2020
135. Han B, Srinivasan G, Roy K, (2020) RMP-SNN: residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. IEEE
136. Tan W, Patel D, Kozma R (2020) Strategy and benchmark for converting deep Q-networks to event-driven spiking neural networks.
137. Deng S, Gu S (2021) Optimal conversion of conventional artificial neural networks to spiking neural networks
138. Yu Q, Ma C, Song S, Zhang G, Dang J, Tan KC (2022) Constructing accurate and efficient deep spiking neural networks with double-threshold and augmented schemes. IEEE Trans Neural Netw Learn Syst 33(4):1714–1726
139. Zhu L, et al (2022) Event-based video reconstruction via potential-assisted spiking neural network
140. Bu T, et al (2022) Optimized potential initialization for low-latency spiking neural networks
141. Meng Q, Yan S, Xiao M, Wang Y, Lin Z, Luo Z-Q (2022) Training much deeper spiking neural networks with a small number of time-steps. Neural Netw 153:254–268
142. Yan Z, Zhou J, Wong W-F (2021) Near lossless transfer learning for spiking neural networks. Association for the Advancement of artificial intelligence(AAAI)
143. Merolla P, Arthur J, Akopyan F, Imam N, Manohar R, Modha DS (2011) A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm. In: Custom Integrated Circuits Conference. pp. 1–4
144. Esser SK, Appuswamy R, Merolla P, Arthur JV, Modha DS (2015) Backpropagation for energy-efficient neuromorphic computing
145. Guo W, et al (2021) Toward the Optimal Design and FPGA Implementation of Spiking Neural Networks. IEEE Transactions on Neural Networks and Learning Systems. PP. 99
146. Mack J, et al (2020) RANC: reconfigurable architecture for neuromorphic computing. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems PP99:1–1
147. Truong-Tuan V, et al (2021) FPGA Implementation of Parallel Neurosynaptic Cores for Neuromorphic Architectures. 2021 19th IEEE International New Circuits and Systems Conference (NEWCAS) IEEE
148. Yang ZT, Han ZJ, Huang YC, Ye TT, IEEE (2021) 55nm CMOS Analog Circuit Implementation of LIF and STDP Functions for Low-Power SNNs. In. IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Electr Network, 2021
149. Zhang J, Liang M, Wei J, Wei S, Chen H, Soc IC (2021) A 28nm configurable asynchronous SNN accelerator with energy-efficient learning. In: 27th IEEE international symposium on asynchronous circuits and systems (ASYNC). Electr Network 2021:34–39
150. Zenke F, Ganguli S (2018) SuperSpike: supervised learning in multilayer spiking neural networks. Neural Comput 30(6):1514–1541
151. She X, Long Y, Mukhopadhyay S (2019) Improving robustness of ReRAM-based spiking neural network accelerator with stochastic spike-timing-dependent-plasticity. IEEE
152. Sharmin S, Panda P, Sarwar SS, Lee C, Ponghiran W, Roy K (2019) A comprehensive analysis on adversarial robustness of spiking neural networks. IEEE

153. Sharmin S, Rathi N, Panda P, Roy K (2020) Inherent adversarial robustness of deep spiking neural networks: effects of discrete input encoding and non-linear activations
154. Kim S, Park S, Na B, Yoon S (2020) Spiking-YOLO: spiking neural network for energy-efficient object detection. Proc AAAI Conf Artificial Intel 34(7):11270–11277
155. Chu H, Yan Y, Gan L, Jia H, Qian L, Huan Y, Zheng L, Zou Z (2022) A neuromorphic processing system with spike-driven SNN processor for wearable ECG classification. IEEE Trans Biomed Circuits Syst 16(4):511–523
156. Mancoo A, Keemink SW, Machens CK (2020) Understanding spiking networks through convex optimization", 34th Conference on Neural Information Processing Systems (NeurIPS 2020)
157. Liang L et al (2021) Exploring adversarial attack in spiking neural networks with spike-compatible gradient, Ieee Transactions on Neural Networks and Learning Systems
158. Kumarasinghe K, Kasabov N, Taylor D (2021) Brain-inspired spiking neural networks for decoding and understanding muscle activity and kinematics from electroencephalography signals during hand movements. Sci Rep 11(1):1–15
159. Lopez-Randulfe J, Duswald T, Bing Z, Knoll A (2021) Spiking neural network for Fourier transform and object detection for automotive radar. Front Neurorobotics, vol. 15, Art no. 688344
160. Kwon D, Jung G, Shin W, Jeong Y, Hong S, Oh S, Kim J, Bae JH, Park BG, Lee JH (2021) Efficient fusion of spiking neural networks and FET-type gas sensors for a fast and reliable artificial olfactory system. Sensors Actuators B Chem 345:130419
161. Wu J, Chua Y, Zhang M, Li G, Li H, Tan KC (2021) A tandem learning rule for effective training and rapid inference of deep spiking neural networks. IEEE Transactions on Neural Networks and Learning Systems
162. Li W, Chen H, Guo J, Zhang Z, Wang Y (2022) Brain-inspired multilayer perceptron with spiking neurons
163. Li Y, et al. (2022) Neuromorphic data augmentation for training spiking neural networks
164. Liu F, et al (n.d.) DynSNN: A Dynamic Approach to Reduce Redundancy in Spiking Neural Networks. ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 0

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.