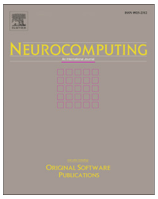
Neurocomputing 531 (2023) 163–179

Contents lists available at ScienceDirect

Neurocomputing

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



Survey paper

Learning rules in spiking neural networks: A survey

Zexiang Yi [a], Jing Lian [b], Qidong Liu [c], Hegui Zhu [d], Dong Liang [e], Jizhao Liu [a,*]

[a] School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, Gansu, China
[b] School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, Gansu, China
[c] School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, Henan, China
[d] College of Sciences, Northeastern University, Shenyang 110819, Liaoning, China
[e] Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, Jiangsu, China

ARTICLE INFO

Article history:
Received 24 October 2022
Revised 11 January 2023
Accepted 11 February 2023
Available online 16 February 2023

Keywords:
Spiking neural networks
Pulse-coupled neural networks
Neuromorphic computing
Learning rules
Image classification

ABSTRACT

Spiking neural networks (SNNs) are a promising energy-efficient alternative to artificial neural networks (ANNs) due to their rich dynamics, capability to process spatiotemporal patterns, and low-power consumption. The complex intrinsic properties of SNNs give rise to a diversity of their learning rules which are essential to functional SNNs. This paper is aimed at presenting a comprehensive overview of learning rules in SNNs. Firstly, we introduce the basic concepts of SNNs and commonly used neuromorphic datasets. Then, guided by a hierarchical classification of SNN learning rules, we present a comprehensive survey of these rules with discussions on their characteristics, advantages, limitations, and performance on several datasets. Moreover, we review practical applications of SNNs, including event-based vision and audio signal processing. Finally, we conclude this survey with a discussion on challenges and promising future research directions in this area.

1. Introduction

The human brain is the best-known intelligent system, performing functions such as perception, reasoning, and control with a power consumption of nearly 20 W [1]. There have been many artificial intelligence (AI) models inspired by the brain. Rosenblatt [2] proposed the perceptron to realize binary classification of input patterns. Lecun et al. [3] applied convolutional neural networks (CNNs) to handwritten digit recognition. Recently, neural networks based on attention mechanisms [4] have invoked a new wave of research. Although ANNs have achieved great success in many fields [5], they have the following drawbacks: 1) the training and inference of ANNs consume huge amounts of energy [6]; 2) limitations such as poor robustness and catastrophic forgetting [7]. However, these problems do not exist in the brain where neurons have complex spatiotemporal dynamics, communicating and processing information through discrete spikes. Additionally, neurons are connected in a hierarchical way, forming different functional neural networks with diverse plasticity. How to introduce the above characteristics to construct more energy-efficient and robust AI models is an open problem in AI research.

1.1. Unique characteristics of spiking neural networks

Spiking neural networks (SNNs) mimic the way the human brain process information by taking advantage of discrete and asynchronous spikes, thus they are believed to have the capability to process spatiotemporal information efficiently [8–10]. The basic building blocks of SNNs are spiking neurons. Due to the description of the generation of spikes at different levels of bio-fidelity, there is a diversity of spiking neuron models such as Hodgkin-Huxley (H-H) model [11], Izhikevich model (IM) [12]. Additionally, by utilizing the time dimension, spiking neurons can represent information in a sparse and robust way [13].

SNNs are bridges between brain science (BS) and AI. On the one hand, neuroscientists use SNNs to simulate biological neural networks to deepen their understanding of the brain [14–16]; On the other hand, AI researchers draw inspiration from BS to build energy-efficient and robust neural networks [17,18]. However, Due to the non-differential nature of spikes, training efficient and high-performance SNNs has remained a major difficulty [19–21].

1.2. Motivation

In recent years, SNNs have attracted enormous research interest. There has been an upward trend in SNN-related papers. To

* Corresponding author.
E-mail address: liujz@lzu.edu.cn (J. Liu).

https://doi.org/10.1016/j.neucom.2023.02.026
0925-2312/© 2023 Elsevier B.V. All rights reserved.

visualize this trend, we present Fig. 1, which illustrates the number of SNN papers that are available on the *Web of Science* since 2015.

Learning is an essential part of SNNs that adapts a network to perform specific tasks, such as classification or object detection. Fig. 2 shows some of the important learning rules over the past two decades. SpikeProp [22] is the earliest spike-based backpropagation in multilayer SNNs. Tempotron [23] can perform binary classification tasks in analogy to perceptron. Remote supervised method (ReSuMe) [24] and spike pattern association neuron (SPAN) [25] are classical spike sequence learning rules. Masquelier and Thorpe [26] apply spike-timing-dependent plasticity (STDP) to multilayer neural networks inspired by ventral visual pathways to enable unsupervised feature learning. In 2015 and beyond, SNNs have been dominated by deep networks. Cao et al. [27] propose converting a pre-trained ANN to an SNN. Kheradpisheh et al. [28] use STDP to train deep spiking CNNs layer by layer. SuperSpike [29] and spatio-temporal backpropagation (STBP) [30] train multilayer SNNs via surrogate gradients. Spike-element-wise (SEW) ResNet [19] is proposed to combat degradation problem in deep SNNs. Bu et al. [31] realize ultra-low-latency inference in ANN-converted SNNs.

Several survey papers [32–41] have so far reviewed recent advances in SNNs. However, some of these papers have a limited scope of learning rules, for instance, [36,34,41] focus on supervised learning, [32,33] have an emphasis on learning rules in multilayer SNNs. Moreover, most of the above surveys only cover the papers published until 2021. Nonetheless, many important breakthroughs in learning rules in SNNs have occurred since 2022. Additionally, none of the surveys cover pulse-coupled neural networks (PCNNs), which are cortex models exhibiting synchronous oscillation behavior and have been widely applied to image processing [42–44].

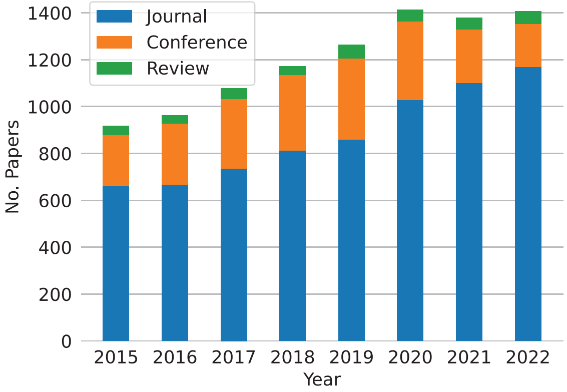


Fig. 1. The number of SNN papers published after 2015.

1.3. Contribution

This paper surveys the advances in learning rules in SNNs, providing insights into both techniques and performance in a systematic way.

The key contributions of this article are summarized as follows. First, we provide a systematic review of the evolution of learning rules in SNNs, where many of them have not been reviewed in previous surveys, and present comparisons between the state-of-the-art using results reported on several popular datasets. Second, we review practical applications of SNNs, including event-based vision and audio signal processing. Third, we discuss a number of challenges and promising future research directions.

1.4. Organization

The rest of this survey is structured as follows. We start by providing the basic concepts of SNNs including neuron and network models, synaptic plasticity, and neural coding. Next, in Section 3, we review the commonly used neuromorphic datasets. Section 4 presents a hierarchical classification of learning rules in SNNs and analyzes the research trend and their characteristics, advantages, limitations, and performance on several datasets. Section 5 reviews practical applications of SNNs. Finally, Section 6 discusses some challenges and directions of this field.

2. Basic concepts of SNNs

SNNs involve more neuroscience-related concepts than ANNs. To better analyze the learning rules in SNNs, basic concepts of SNNs are introduced in this section, including neuron and network models, synaptic plasticity, and neural coding.

2.1. Neuron models

There are billions of neurons in the human brain which have a basic structure as shown in Fig. 3. Dendrite is the input terminal. The cell body integrates incoming spikes received by different branches of dendrites and emits a spike when its membrane potential reaches the threshold. Spikes travel along the axons to other neurons via synapses.

To emulate the generation of spikes with different levels of biofidelity and computational cost, a variety of spiking neuron models have been proposed. For simplicity and mathematical tractability, leaky integrate-and-fire (LIF) [45], spike response model (SRM) [46] and PCNN neuron [14] models are widely used in SNNs. To better formalize these models, Table 1 summarizes the main notations used in the following equations.

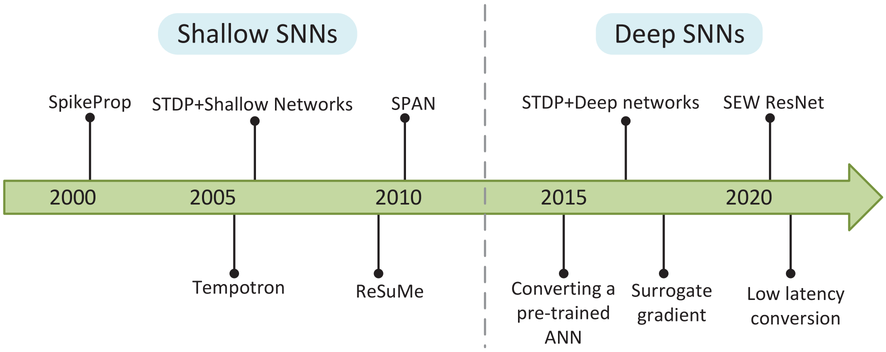


Fig. 2. The evolution of learning rules in SNNs.

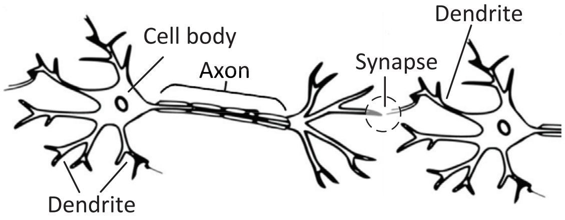


Fig. 3. Diagram of a neuron. It can be divided into three parts: dendrite, cell body, and axon.

Table 1
Notation list.

Notation	Description
τ, R, C	Time constant, input resistance and capacitor of a neuron, respectively
F	Feeding input of a neuron
L	Linking input of a neuron
V	Membrane potential of a neuron
E	Dynamic threshold of a neuron
S	Output spike train of a neuron
t_s^j	Time of jth spike of output spike train
S_i	Input spike train from ith synapse
t_i^j	Time of jth spike from ith synapse

LIF model dates to 1907 [45], when the mechanism of generating spikes had not yet been revealed, so neurons were modeled as a parallel circuit of resistance R and a capacitance C as shown in Fig. 4a. When the input current I is injected into the capacitor, its voltage V rises. Meanwhile, charges on the capacitor will leak through the resistor. A spike will be emitted whenever V reaches the threshold V_{th}, which is then reset to the resting voltage V_{rest}. Fig. 4b depicts the dynamics of a LIF neuron under constant input, which can formally be described in differential form as:

$$\tau_m \frac{dV}{dt} = -(V - V_{rest}) + RI + S(t)(V_{rest} - V_{th}) \tag{1}$$

$$S(t) = \sum_{t_j \leqslant t} \delta(t - t_s^j) \tag{2}$$

where $\tau_m = RC$ is the time constant of the membrane. When working with differential equations, it is convenient to denote a spike as a Dirac delta function $\delta(t)$, so the postsynaptic spike train $S(t)$ can be presented as a sum of Dirac functions at different output spike times t_s^j.

Synapse transmits spikes via neurotransmitter which acts like a low pass filter with synaptic weight. Dendrite integrates the weighted and filtered synaptic current, obtaining the total input current I. The dynamics of these operations are given by:

$$\tau_s \frac{dI}{dt} = -I + \sum_i w_i s_i(t) \tag{3}$$

$$S_i(t) = \sum_{t_i \leqslant t} \delta\left(t - t_i^j\right) \tag{4}$$

where τ_s is the synapse time constant. t_i^j denote the presynaptic spike times of ith afferent. The sum runs over all presynaptic neurons i. w_i and $s_i(t)$ are the corresponding synaptic weights and presynaptic spike trains, respectively.

It is customary to simulate SNNs in discrete time using Euler's method. To reduce computational costs, the synaptic filter effect is often ignored, but there are also researchers [47] incorporating this filter dynamics to improve the convergence of SNNs training. In addition, there are some variants of the LIF Model such as quadratic integrate-and-fire (QIF) and integrate-and-fire (IF) models.

SRM uses spike response kernels to model a neuron's membrane potential in response to its input and out spikes [46]. A commonly used form is given as follows:

$$V(t) = \sum_i w_i \sum_{t_i \leqslant t} \varepsilon\left(t - t_i^j\right) + \sum_{t_j \leqslant t} \eta(t - t_s^j) + V_{rest} \tag{5}$$

where ε and η are the input and output spike response kernel, respectively.

SRM neuron models are appealing as they can add other features simply by embedding them into the kernel. In addition, since the membrane potential is explicitly expressed, the simulation of SRM models is often event-based, thus less cost and time-consuming than LIF models.

The PCNN neuron is a two-compartmental model [42], as shown in Fig. 5. The dendrite tree has two distinct inputs, the primary input termed feeding input F, and the auxiliary input termed linking input L. They are represented as leaky integrator given by

$$\tau_f \frac{dF}{dt} = -F + I_f \tag{6}$$

$$\tau_l \frac{dL}{dt} = -L + I_l \tag{7}$$

where τ_f, τ_f, I_f and I_l are the time constants and synaptic currents of the feeding and linking input, respectively. The synaptic currents

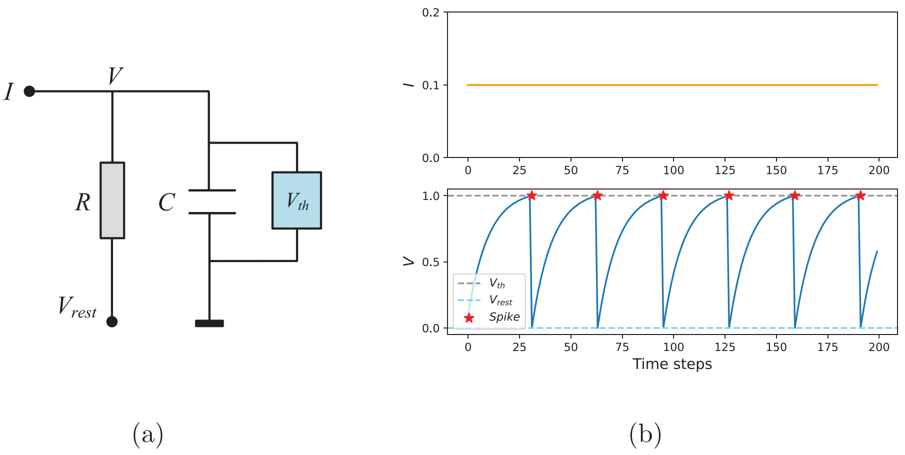


Fig. 4. The LIF model. (a) Diagram of the LIF model circuits. (b) Neuronal dynamics of a LIF neuron under the constant current.

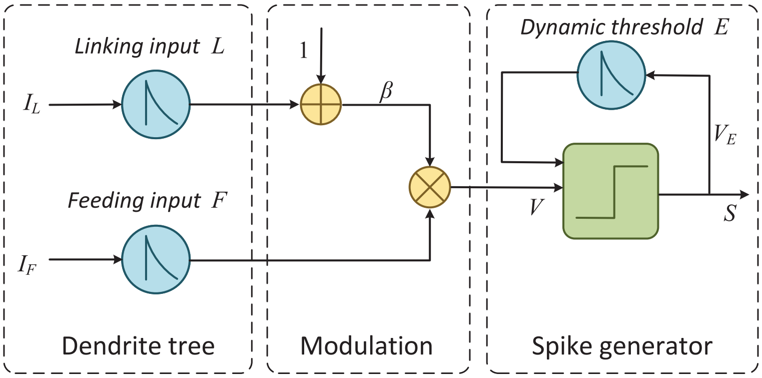


Fig. 5. Schematic of a PCNN neuron. It consists of three parts: dendrite tree, modulation, and spike generator.

can be spikes, constants, analog time-varying signals, or any combination.

The linking input modulates the feeding input by multiplication resulting in the membrane potential V.

$$V = F(1 + \beta L) \tag{8}$$

where β is the linking strength.

The spike generator will emit a spike whenever the membrane potential crosses the threshold E. Unlike LIF models resetting the membrane potential, the PCNN neuron feeds back to the threshold, which is another leaky integrator given by

$$\tau_e \frac{dE}{dt} = -E + V_E S(t) \tag{9}$$

where $S(t)$ denotes the output spike train of the PCNN neuron. τ_e is the time constant and V_E is the amplitude gain.

2.2. Network models

Neurons are connected to form neural networks following some connection patterns, as shown in Fig. 6. There are three types of feedforward connections, convolutional, local, and full connection. Convolutional and local connections both have local visual receptive fields, but in convolutional connection, the weights are shared between all receptive fields, while in local connection, each receptive field has its own set of weights, which is more biologically plausible [48]. Full connection layers are often used to classify extracted features. Recurrent connections consist of self-recurrent, lateral, and feedback connections. Zhang and Li [49] added self-recurrent connections to implement local memory. Diehl and Cook [50] used feedback connections to implement winner-take-all mechanism. Cheng et al. [17] introduced lateral connections to improve the recognition robustness against noise. The last type of connection is the residual connection which can solve the degradation problem in deep ANNs and is also the key to deep SNNs [19,20].

Network models can be constructed using the above connection patterns. We review three types of network models as follows. The PCNN is a cortex model originated from the simulation of synchronous oscillation behavior in the primary cortex of cats [42]. The classical PCNN and its variants such as the spiking cortical model (SCM) [51] have been widely used in image segmentation [52,53], fusion [54–56], enhancement [57–59], invariant texture retrieval [60], and many other image processing tasks [61–63]. Although a vast body of works utilize PCNNs for image processing, less attention has been paid to learning rules for PCNNs as we will see in the following. Based on the characteristics of ventral visual flow, Riesenhuber and Poggio[64] proposed the HMAX model. Serre et al. [65] expanded HMAX into the field of computer vision. Masquelier and Thorpe Masquelier and Thorpe [26] proposed a spiking version of HMAX equipped with STDP for unsupervised feature learning. Single layer networks (SLNs), multilayer Perceptron (MLP), recurrent neural networks (RNNs), and Convolutional neural networks (CNNs) are network models widely used in deep learning [66]. VGG and ResNet are popular neural architectures for deeper CNNs which have been adopted by a large number of high-performance SNNs.

2.3. Synaptic plasticity

Synaptic plasticity refers to the modulation of synaptic weights [67]. In 1949, Hebb [68] proposed Hebb's postulate, and it can be simply stated as ”neurons that fire together, wire together [69]. ” Subsequent finding of long-term potentiation [70] provided experimental evidence for Hebb's postulate. LTP together with long-term depression (LTD) regulates synaptic weights bidirectionally, serving as the synaptic basis for learning and memory. The induction of LTP and LTD is spike-timing dependent. Studies [71,72] demonstrated that the relative timing of the pre and postsynaptic spikes

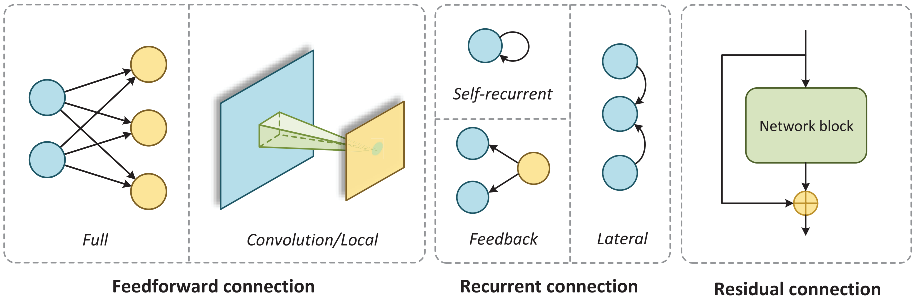


Fig. 6. Connection patterns in SNNs.

determine the direction and magnitude of synaptic modification. This phenomenon is known as spike-timing-dependent plasticity (STDP) [73]. Fig. 7a shows two neurons connected by a synapse, and Fig. 7b illustrates the relationship between the amount of change in synaptic weights and the timing of pre and postsynaptic spikes.

Fitting experimental data with exponential function, we can formalize STDP as

$$\Delta w = \begin{cases} A_+ \exp\left(\frac{-\Delta t}{\tau_+}\right) & \Delta t > 0 \\ -A_- \exp\left(\frac{\Delta t}{\tau_-}\right) & \Delta t < 0 \end{cases} \tag{10}$$

where A_+ and A_- are the modulation magnitudes for LTP and LTD, respectively. τ_+ and τ_- are the corresponding time constants of the learning window, and $\Delta t = t_{post} - t_{pre}$ is the time difference between a pair of pre and postsynaptic spikes. In STDP modeling studies, Song et al. [73] found that synapses modified by STDP compete with one other, resulting in a bimodal distribution of synaptic weights. Guyonneau et al.[74] showed that a neuron with STDP-modified synapse stimulated by a repeatedly presented spike pattern will be selective to it and decrease response latency. STDP also enables neurons to learn visual features in an unsupervised way [26,50,75].

Increasing experimental observations demonstrate that neuromodulators play a vital role in synaptic plasticity. They can change the polarity [76] or adjust the time window of STDP [77]. Frémaux and Gerstner [78] proposed three-factor rules to incorporate the influence of neuromodulators. Inspired by these observations, researchers model neuromodulator effects to implement bio-plausible supervised learning [79] and reinforcement learning [80] for image recognition.

2.4. Neural coding

Stimulus, such as light or odors are converted to spikes for neural processing, a process known as neural coding. Currently, there are three main coding methods applied in SNNs: rate, temporal and direct coding.

Rate coding converts input stimulus into Poisson-distributed spike trains, with firing rates proportional to the input intensity. To reduce the computational cost, the binomial distribution is commonly used instead of Poisson distribution [17].

As rate coding represents information via firing rate, i.e. spike count over a short time, a single spike contains little information. temporal coding is concerned with the timing of a spike. A common temporal coding method is time-to-first-spike (TTFS) coding [81] in which a larger input intensity corresponds to an earlier spike. Therefore, TTFS coding requires fewer spikes to encode information than rate coding, resulting in a lower power consumption and inference latency [82].

Rueckauer et al. [83] suggested that the variability in rate coding impairs the performance of SNNs. So many studies [84,85] used a trainable spiking neuron layer to convert analog input that can be regarded as synaptic currents into output spike strains, which is called direct coding [86].

3. Neuromorphic datasets

The performances of SNNs are often evaluated on existing ANN-oriented datasets, for example CIFAR-10 [87] and ImageNet [88], which are static-image datasets containing no temporal information. Before feeding them into SNNs, researchers often convert such frame-based data to spike trains using coding methods described in Section 2.4. However, these ANN-oriented datasets can't exploit the spatiotemporal processing capability of SNNs [89]. To this end, researchers have gathered neuromorphic datasets inspired by the biological visual system.

Neuromorphic datasets are recorded by dynamic vision sensors (DVS) which capture the changes in the sensing field using two channels. The On channel for intensity increases and the Off channel for intensity decreases. Currently, neuromorphic datasets can be divided into two categories: DVS-converted and DVS-captured [90]. DVS-converted datasets are converted from traditional datasets, such as MNIST and CIFAR-10. Researchers use a DVS camera to record static images by moving the image or camera. Both N-MNIST [91] and CIFAR10-DVS [92] are acquired by this method. In contrast, DVS-captured datasets are recorded via real-world motion. DVS128 Gesture [93] is a typical example. We present an overview of the main characteristics of well-known neuromorphic datasets in Table 2. In the following, we review these datasets in detail.

N-MNIST: The N-MNIST dataset [91] is converted from the MNIST dataset. Researchers first displayed MNIST examples on an LCD monitor, and then moved the ATIS sensor mounted on a pan-tilt unit to record the image. It includes a training set with 60,000 samples and a test set with 10,000 samples.

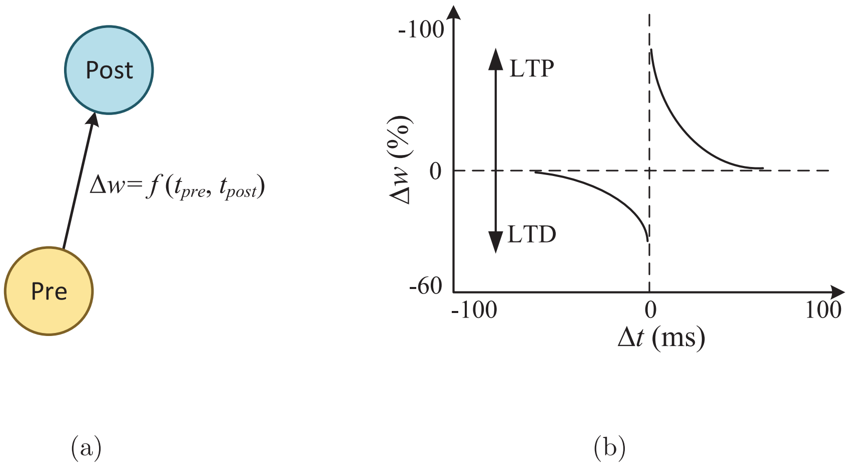


Fig. 7. STDP. (a) Two neurons are connected by a synapse equipped with STDP. (b) Relationship between spike timing and synaptic weight change. In a time window of tens of milliseconds, when the presynaptic spike is earlier (later) than the postsynaptic spike, the weight increases (decreases), resulting in LTP (LTD).

Table 2
Summary of well-known neuromorphic datasets.

Datase	Year	Data category	#classes	URL
N-MNIST	2015	DVS-converted	10	http://www.garrickorchard.com/datasets
N-Caltech101	2015	DVS-converted	101	http://www.garrickorchard.com/datasets
CIFAR10-DVS	2017	DVS-converted	10	https://figshare.com/s/d03a91081824536f12a8
DVS128 Gesture	2019	DVS-captured	11	http://research.ibm.com/dvsgesture
ASL-DVS	2019	DVS-captured	24	https://github.com/PIX2NVS/NVS2Graph

N-Caltech101: The N-Caltech101 dataset [91] is a neuromorphic version of the N-Caltech101 dataset. It was recorded using the same method as the N-MNIST dataset and contains 9146 images in 101 classes.

CIFAR10-DVS: The CIFAR10-DVS dataset [92] is a spiking version of the CIFAR-10 dataset. The dataset was recorded by moving images in front of a DVS camera. It consists of 10,000 examples in 10 classes, with 1000 examples in each class.

DVS128 Gesture: The DVS128 Gesture dataset [93] was recorded by a DVS128 camera and contained 11 kinds of hand gestures from 29 subjects under 3 kinds of illumination conditions.

ASL-DVS: The ASL-VDS dataset [94] was recorded in an office environment with low environmental noise and constant illumination. It contains 24 classes of gestures corresponding to 24 English letters.

4. Learning rules in SNNs

In this section, we first introduce the hierarchical classification of SNN learning rules which gives an overview of these rules. Then, we analyze the research trends in SNN learning rules. Finally, we review these rules in a hierarchical way and summarize the state-of-the-art performance for comparison on several datasets.

4.1. Hierarchical classification of SNN learning rules

In order to help illustrate an overall structure for SNN learning rules, we categorized the existing learning rules hierarchically, which is presented in Fig. 8. Previous papers [35,95] classify SNN learning rules according to the usage of data label, i.e. supervised and unsupervised learning [35] or the biological realism and plas-

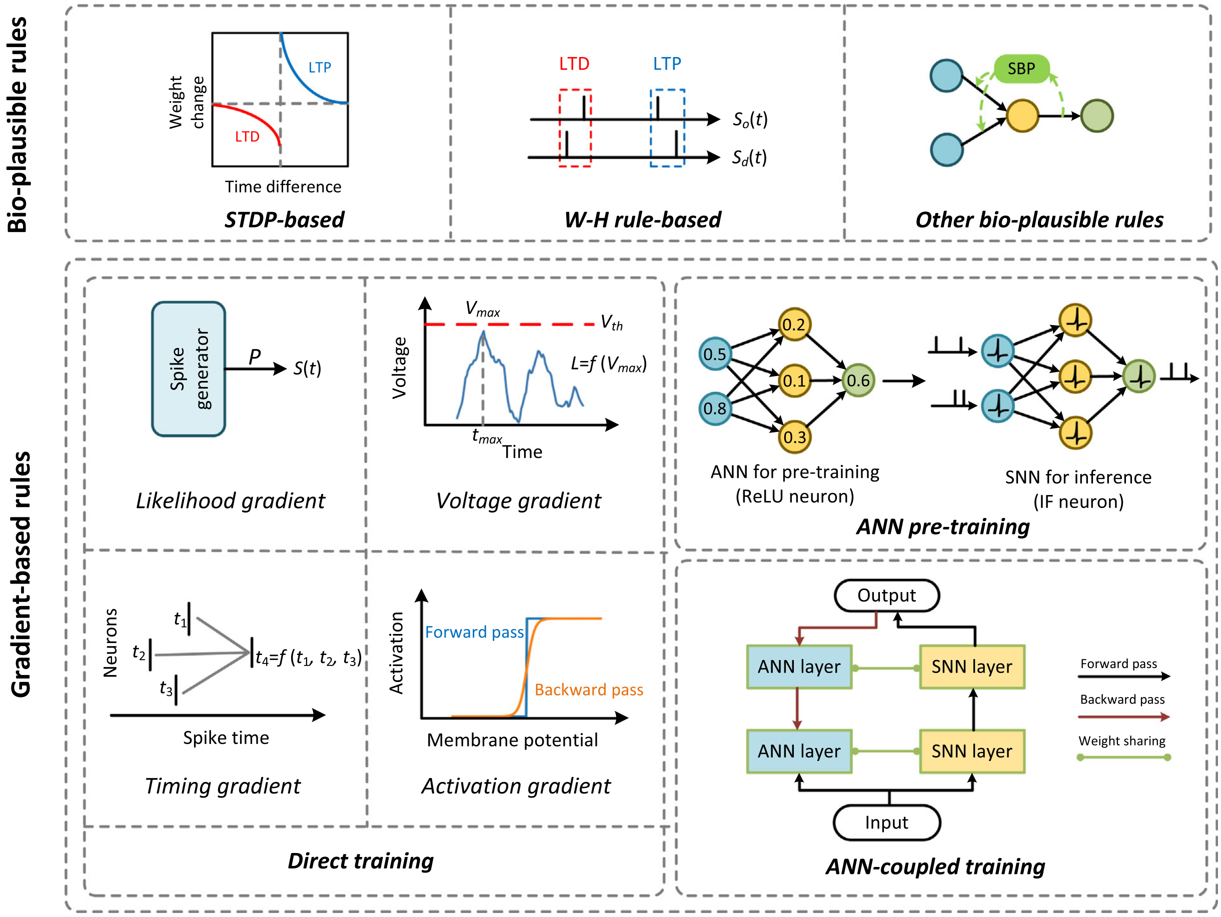


Fig. 8. Illustration of the three-level hierarchical classification of learning rules in SNNs. Bold only, bold italic, and italic only style means the first, second, and third level of the hierarchy, respectively.

ticity scale of learning rules [95]. We differ from these papers by considering the working principles of the SNN learning rules. The details of each group of learning rules will be discussed later.

4.2. Analysis and trends

Before we dive into individual rules, we summarize the most representative SNN learning rules in Table 3, sorted according to their publication dates, for analysis of the evolution and trends in this field. LIF and SRM are mainly used neuron models for their simplicity and mathematical tractability. Direct coding has become popular due to its advantages over rate coding in accuracy [86]. Temporal coding is recently limited to certain rules such as timing gradient. Notably, there has been a clear trend in formulating efficient learning rules such as ANN pre-training and activation gradient for deeper SNNs ($>$ 10 layers) to perform challenging tasks, such as image classification on ImageNet and CIFAR-100.

4.3. Bio-plausible rules

Driven by considerations of biological plausibility, bio-plausible rules primarily focus on implementation via experimentally observed biological phenomena such as STDP, coherent oscillation [126], and self-backpropagation [127]. We classify these rules into STDP-based, Widrow-Hoff (W-H) rule-based [128] and other bio-plausible rules according to the underlying principles they are derived from.

Table 3
Summary of some representative learning rules in SNNs

Ref.	Year	Venue	Neuron Model	Coding Method	Learning Rule	Network Model	Dataset
[22]	2002	*Neurocomp.*	SRM	Temporal	Timing gradient	MLP	Iris
[23]	2006	*Nat. Neurosci.*	LIF	–	Voltage gradient	SLN	–
[96]	2006	*Neural Comput.*	SRM	–	Likelihood gradient	SLN	–
[26]	2007	*PLoS Comput. Biol.*	IF	Temporal	STDP-based	HMAX	Caltech-101
[97]	2009	*Neural Netw.*	SRM	Temporal	Timing gradient	MLP	Iris
[24]	2010	*Neural Comput.*	LIF; H-H; IM	–	W-H rule-based	SLN	–
[98]	2012	*PLoS One*	SRM	–	W-H rule-based	SLN	–
[99]	2013	*PLoS One*	LIF; IM	–	W-H rule-based	SLN	–
[100]	2014	*Neurocomp.*	SRM	Temporal	STDP-based	MLP	Iris
[50]	2015	*Front. Comput. Neurosci.*	LIF	Rate	STDP-based	SLN	MNIST
[101]	2015	*Neural Comput.*	SRM	–	Likelihood gradient	MLP	–
[27]	2015	*IJCV*	IF	Rate	ANN pre-training	CNN	CIFAR-10; Neovision2
[102]	2015	*IJCNN*	IF	Rate	ANN pre-training	CNN	MNIST
[103]	2015	*IJCNN*	LIF	–	Voltage gradient	SLN	–
[104]	2016	*Science*	LIF	–	Voltage gradient	SLN	TIDIGITS
[105]	2016	*arXiv*	IF	Direct	ANN pre-training	CNN	CIFAR-10
[106]	2016	*Neurocomp.*	IF	Temporal	STDP-based	HMAX	3D-Object
[107]	2016	*Front. Neurosci.*	LIF	Rate	Voltage gradient	CNN	MNIST; N-MNIST
[83]	2017	*Front. Neurosci.*	LIF	Direct	ANN pre-training	VGG-16	MNIST; CIFAR-10; ImageNet
[108]	2018	*IEEE T-NNLS*	IF	Temporal	Timing gradient	MPL	MNIST
[109]	2018	*NeurIPS*	IF; QIF	–	Activation gradient	RNN	–
[28]	2018	*Neural Netw.*	IF	Temporal	STDP-based	CNN	Caltech-101; ETH-80; MNIST
[80]	2018	*IEEE T-NNLS*	IF	Temporal	STDP-based	HMAX	Caltech-101; ETH-80;
[110]	2018	*Neural Netw.*	LIF	Direct	Voltage gradient	CNN	MNIST
[30]	2018	*Front. Neurosci.*	LIF	Rate	Activation gradient	CNN	MNIST; N-MNIST
[85]	2019	*AAAI*	LIF	Direct	Activation gradient	CNN	N-MNIST; CIFAR10-DVS; CIFAR-10
[111]	2019	*IEEE T-Cyb.*	LIF	–	Voltage gradient	SLN	–
[112]	2019	*IEEE T-CDS*	LIF	Rate	STDP-based	CNN	Caltech-101; MNIST
[48]	2019	*Neural Netw.*	LIF	Rate	STDP-based	SLN	MNIST
[113]	2019	*Front. Neurosci.*	IF	Rate	ANN pre-training	VGG-16; ResNet-20/34	CIFAR-10; ImageNet
[82]	2020	*Int. J. Neural Syst.*	IF	Temporal	Timing gradient	MLP	Caltech-101; MNIST
[114]	2020	*ICASSP*	SRM	Temporal	Timing gradient	MLP	MNIST
[115]	2020	*CVPR*	IF	Rate	ANN pre-training	VGG-16; ResNet-20/34	CIFAR-10; CIFAR-100; ImageNet
[116]	2020	*ICLR*	IF	Rate	ANN pre-training	VGG-16; ResNet-20/34	CIFAR-10; CIFAR-100; ImageNet
[117]	2021	*IEEE T-NNLS*	IF	Direct	ANN-coupled training	CifarNet; AlexNet	CIFAR-10; ImageNet
[17]	2021	*IJCAI*	LIF	Rate	Activation gradient	CNN	MNIST; Fashion-MNIST
[84]	2021	*ICCV*	LIF	Direct	Activation gradient	CNN	CIFAR10; CIFAR10-DVS; DVS128 Gesture
[118]	2021	*AAAI*	IF	Temporal	Timing gradient	VGG-16; GoogleNet	MNIST; CIFAR-10; ImageNet
[119]	2021	*ICLR*	IF	Direct	ANN pre-training	VGG-16; ResNet-20	CIFAR-10; CIFAR-100; ImageNet
[120]	2021	*ICML*	IF	Direct	ANN pre-training	VGG-16; ResNet-20/34; RegNetX	CIFAR-10; CIFAR-100; ImageNet
[121]	2021	*IEEE T-NNLS*	IF	Direct	ANN pre-training	ResNet-50/110	CIFAR-10; CIFAR-100; ImageNet
[122]	2021	*IJCAI*	IF	Direct	ANN pre-training	VGG-16; PreActResNet-18/34	MNIST; CIFAR-10; CIFAR-100
[123]	2021	*Sci. Adv.*	LIF	Rate	Other bio-plausible rule	MLP	MNIST; NETtalk; DVS128 Gesture
[19]	2021	*NeurIPS*	LIF; IF	Direct	Activation gradient	ResNet-18/34/50/101/152	ImageNet; DVS128 Gesture; CIFAR10-DVS
[20]	2021	*arXiv*	LIF	Direct	Activation gradient	ResNet-104/482	CIFAR10-DVS; ImageNet
[124]	2022	*AAAI*	IF	Direct	ANN pre-training	VGG-16; ResNet-18/20	CIFAR-10; CIFAR-100; ImageNet
[31]	2022	*ICLR*	IF	Direct	ANN pre-training	VGG-16; ResNet-18/20/34	CIFAR-10; CIFAR-100; ImageNet
[125]	2022	*ICLR*	LIF	Direct	Activation gradient	VGG-11;ResNet-19/34	CIFAR-100; ImageNet; CIFAR10-DVS

4.3.1. STDP-based rules

As mentioned in Section 2.3, STDP observed in biology experiments can enable a neuron to learn visual features in an unsupervised way. Thus STDP-based rules have been studied in many research. Masquelier and Thorpe [26] proposed an HMAX-based SNN trained with unsupervised STDP to extract visual features from a temporal coded image. Many subsequent works were based on this method. Tavanaei and Maida[129] incorporated probabilistic STDP to improve performance. Kheradpisheh et al. [106] expanded this method to perform robust invariant object recognition tasks. Unsupervised STDP can extract repeated features; however, it has difficulty in detecting rare but diagnostic features. To this end, Mozafari et al. [80] introduce reward signals to STDP, called reward-modulated STDP (R-STDP), to improve feature extraction ability.

There are several ways to formalize supervised STDP. Wang et al. [100] combined STDP and anti-STDP to implement supervised learning for the output layer of a two-layer SNN. When a neuron emits spikes correctly, STDP is applied, otherwise, anti-STDP is applied. Beyeler et al. [130] used supervisory neurons to send excitatory signals to target output neurons, making them spike at the desired firing rate. Illing et al. [131] implemented supervised learning via target post spike trace. Synaptic weights update at every presynaptic spike times. When the actual post spike trace is lower than the target value, the corresponding weight increases, and decreases instead. Hao et al. [79] introduced dopamine-modulated STDP (DA-STDP) combined with synaptic scaling to realize supervised learning.

As early works to perform digit recognition on MNIST using unsupervised STDP, Querlioz et al. [132,133] proposed a single-layer network with lateral inhibition and dynamic threshold, yielding an accuracy of 93.5% with 300 neurons. They used memristors as synapses. Although the rectangular STDP time-window was used for modulation of synaptic weights in these works, the memristive devices can implement biological STDP learning rule easily [134,135]. Diehl and Cook [50] increased the network scale and improved its biological plausibility, yielding an accuracy of 95%. It was further improved by Saunders et al. [48,136] with local connections, resulting in reduced parameters and training time. To take advantage of the feature extraction capabilities of CNNs, Xu et al. [137] proposed deep CovDenseSNN which uses spiking neurons to learn features extracted by CNNs.

Recent works adopted CNN-like neural architecture for SNNs. Kheradpisheh et al. [28] used STDP to train a three-layer spiking CNN layer-by-layer. The complexity of learned features increases along the network hierarchy. A support vector machine (SVM) was used for feature classification and achieved an accuracy of 98.6% on MNIST. Mozafari et al. [138] improved the biological plausibility of this model by placing SVM with a layer of decision-making neurons trained with R-STDP. Unlike previous works, SpiCNN [112] used rate coding with Poisson distribution and an output layer trained with supervised STDP.

4.3.2. W-H rule-based rules

The W-H rule [128] is a classical learning rule proposed for ANN neurons defined as

$$\Delta w_i = \eta x_i (y_d - y_o) \tag{11}$$

where x_i is the input of the ith synapse. y_o and y_d are the actual and desired output of the neuron, respectively. η is the learning rate. The W-H rule requires no need for gradient calculation, so there is a bunch of SNN learning rules that present a spiking analogy to this rule.

ReSuMe [24] interprets the W-H rule through two biological processes: STDP and anti-STDP, which is illustrated in Fig. 9. It updates the synaptic weights according to

$$\frac{dw_i(t)}{dt} = [s_d(t) - s_o(t)]\left[a + \int_0^{+\infty} T(s) s_i(t-s) ds\right] \tag{12}$$

where $s_d(t)$ and $s_o(t)$ are the desired and actual output spike trains, respectively. a is a constant used for speeding up the convergence of learning. $T(s)$ is the STDP-like learning window. The convolution of the learning window and the ith input spike train $\int_0^{+\infty} T(s) s_i(t-s) ds$ represents the trace of the spike train.

Unlike the spike-driven update of the synaptic weights in ReSuMe, the perceptron-based spiking neuron learning rule (PBSNLR) [139] transforms a spiking neuron to a perceptron at fixed points in time, where the membrane potential in the misclassification intervals is utilized for synaptic updating. Inspired by the biological property that the synaptic delay is not constant, Taherkhani et al. [140] proposed a synaptic delay learning rule, delay-learning ReSuMe (DL-ReSuMe), which can improve the learning accuracy and convergence speed. To overcome the one-way adjustment problem in DL-ReSuMe, Zhang et al. [141] proposed synaptic weight-delay plasticity for ReSuMe (ReSuMe-DW) and PBSNLR (PBSNLR-DW) that can decrease the delay to increase the membrane potential at desired spike times.

Inspired by ReSuMe, the chronotron I-learning [98] is a heuristic learning rule, which adjusts the synaptic weights proportionally to their corresponding synaptic currents. SPAN [25] uses convolving kernels to convert input, actual, and desired spike trains into analog signals. Unlike SPAN, precise-spike-driven (PSD) synaptic plasticity [99] only convolves input spike trains.

These rules make neurons fire spikes at desired times. By emitting different spike trains, neurons can classify input spike patterns. However, they do not extend well to deep networks which are vital for solving complex tasks.

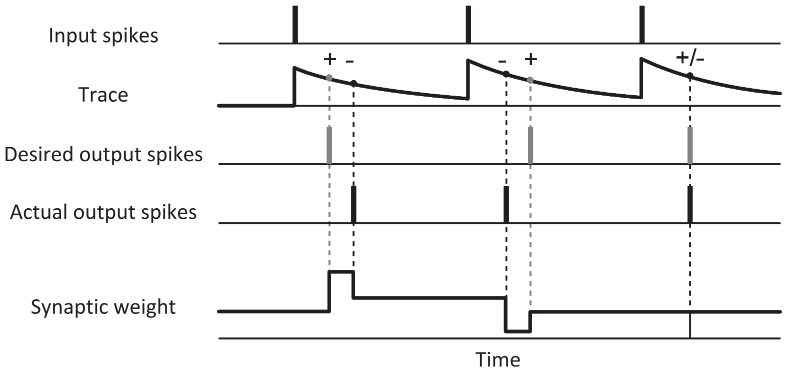


Fig. 9. Illustration of the ReSuMe rule [24]. The amount of modification of an excitatory synaptic weight is proportional to the trance induced by the convolution of input spikes and the learning window. The STDP process is triggered by the desired output spikes, while the anti-STDP process is triggered by the actual output spikes.

4.3.3. Other bio-plausible rules

STDP modifies synaptic weights depending on the pre and postsynaptic neuronal activity, which lacks global teaching signals for the whole network. To this end, researchers use neural activity-encoded errors to drive synaptic changes that can be backpropagated to upstream layers. Zhang et al. [123] introduced self-backpropagation (SBP) into a three-layer SNN to reduce the computational cost without affecting accuracy. Payeur et al. [142] proposed burst-dependent synaptic plasticity to achieve hierarchical credit assignment in multi-layer SNNs.

Xie et al. [143] found that STDP cannot be applied to recurrent synaptic connection in PCNNs, which obstruct the learning in PCNNs. Inspired by the neural activity-dependent property of STDP, they proposed spike-synchronization-dependent plasticity (SSDP) rule to improve the spike synchronization. Experimental results showed that SSDP-based PCNNs can get better segmentation performance. In addition, they designed a novel memristor-based circuit model of SSDP.

4.4. Gradient-based rules

Application of powerful gradient descent formalism to SNNs is complicated by the hard non-linearity of spike generation mechanism: small changes in synaptic weights can cause large changes in the output spike train, i.e. spikes times or counts. Specifically, the gradient of the output spike train with respect to the synaptic weights $\nabla_w S(t)$ is zero except at spikes times where it is ill-defined [29]. According to the insights on how to circumvent the problem, gradient-based rules can be classified into three categories, direct training, ANN pre-training, and ANN-coupled training.

4.4.1. Direct training

Direct training approaches can be categorized into likelihood, voltage, timing, and activation gradient approaches according to the state variable used for the optimization of the objective function.

Likelihood gradient The threshold nonlinearity can be smoothed via stochasticity which makes it possible to perform gradient descent to maximize the likelihood of generating desired output spike trains. There are various methods to introduce stochasticity, such as stochastic threshold [96] or synapse [144] Pfister et al. [96] used likelihood gradient to train single-layer networks with temporal coding. This approach recently has been extended to multilayer networks [145,101,144,146]. However, it has not been applied to deep networks for complex tasks.

Voltage gradient The voltage of a neuron at time instants is differentiable with respect to its synaptic weights under certain conditions or approximations which facilitates the formulation of gradient-base rules.

Tempotron [23] learns to classify binary spike patterns by minimizing the distance between the firing threshold and shunted membrane potential maximal on error trials. It changes the synaptic weights according to

$$\Delta w_i = \eta(y - \hat{y}) \sum_{t_i^j < t_{max}} K\left(t_{max} - t_i^j\right) \tag{13}$$

where η is the learning rate. t_{max} denotes the time of maximal membrane potential value. t_i^j denotes the jth spike time of the ith synapse. $K(t)$ is the normalized postsynaptic potential (PSP), which is a double exponential. $y \in \{0, 1\}$ is the label of input patterns. $\hat{y}$ denotes the prediction of the neuron, which is determined by

$$\hat{y} = \begin{cases} 1 & V(t_{max}) \geqslant V_{th} \\ 0 & V(t_{max}) < V_{th} \end{cases} \tag{14}$$

The limitation of tempotron to binary classifications was overcome by multi-spike tempotron (MST) [104] which solves multi-classification problems by mapping input spike patterns to desired numbers of output spikes. MST introduces the spike-threshold-surface (STS) function, which maps critical thresholds to the numbers of emitted spikes, to formulate a continuous objective function differentiable with respect to neuron's synaptic weights.

To reduce the computational costs of MST, Yu et al. [111] utilized the linear assumption for threshold crossing [22] to derive the efficient threshold-driven plasticity (TDP) algorithm. Subsequent efficient multispike learning (EML) [147], joint weight-delay plasticity (TDP-DL) [148] further improve MST from various perspectives. Unlike the rules in Section 4.3.2, these multi-spike learning rules can train a neuron to emit a desired number of spikes which is proportional to the number of underlying clues, without specifying the spike times. Thus this type of learning is termed aggregate-label learning [104]. In contrast to MST and TDP, membrane-potential driven aggregate-label learning (MPD-AL) [149] constructs the error functions based on the membrane potential instead of the critical thresholds.

The above-mentioned rules are limited to single neurons, which hinders their practical applications. To this end, Zhang et al. [199] extended aggregate-label learning to multilayer SNNs for multimodal pattern recognition. An alternative approach is encoding static images into sparse spike representations via non-spiking methods, such as S1C1-SNN [150], SCNN [151], CNN-TDP [152] and UMP-TDP [152]. This approach can achieve comparable performance to deep SNNs on small-scale datasets, such as MNIST and Fashion-MNIST.

Normalized approximate descent (NormAD) [103] was derived under the consumption of sparse spike trains, which approximates the gradient of membrane potential at a given time instant with respect to synaptic weights. This approximation method has been effectively applied to multilayer SNNs with a weight-fixed convolutional layer [110], and spiking convolutional auto-encoders which yield 99.08% accuracy on MNIST dataset [153]. Similarly, Zhang et al. [95,154] force the neuron to reset only at desired output spike times [139,155] thus enabling easy calculation of voltage gradients. Unlike obtaining voltage gradient at specific times, Lee et al. [107] ignored discontinuities of membrane potential at spike times and treated the output of a neuron as a linear function of its inputs which has filtered by the membrane. They trained a spiking CNN, achieving an accuracy of 99.31% on MNIST.

Timing gradient The timing gradient approaches focus on the neuron's output spike times and compute the gradients with respect to the neuron's synaptic weights. Due to the event-driven nature of spikes, these approaches allow for efficient even-based network simulation.

SpikeProp [22] is the first algorithm applying backpropagation to temporally coded networks of spiking neurons firing at most one. Its formula is as follows:

$$\Delta w_{ij}^l = -\eta \frac{\partial L}{\partial t_j^l} \frac{\partial t_j^l}{\partial V_j^l(t_j^l)} \frac{\partial V_j^l(t_j^l)}{\partial w_{ij}^l} \tag{15}$$

where η is the learning rate, a positive constant. L is the loss function. V_j^l and t_j^l denote the membrane potential and spike of neuron j in layer l, respectively. w_{ij}^l is the synaptic weight from neuron i in $l - 1$ layer to neuron j in l layer. The key challenge is to solve the partial derivative $\partial t_j^l / \partial V_j^l(t_j^l)$ in Eq. 15. SpikeProp assumes that the membrane potential of a neuron increases linearly in a small enough region around the firing time as shown in Fig. 10. Thus, $\partial t_j^l / \partial V_j^l(t_j^l)$ can be expressed as

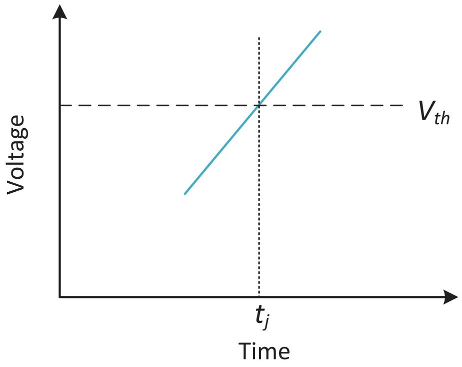


Fig. 10. Illustration of linear approximation for threshold crossing [22]. It assumes that the membrane potential of a neuron increases linearly in a small enough region around the firing time. The sold line denotes the membrane potential when threshold crossing.

$$\frac{\partial t_j^l}{\partial V_j^l(t_j^l)} = \frac{-1}{\partial V_j^l(t_j^l)/\partial t_j^l} = \frac{-1}{\sum_i w_{ij}^l \frac{\partial K(t_j^l - t_i^{l-1})}{\partial t_j^l}} \tag{16}$$

where $K(t)$ is used to describe the PSP of neuron j generated by the input spike t_i^{l-1} , which also denotes the output spike time of neuron i in layer $l-1$.

Multi-SpikeProp [97] overcomes the limitation of one spike in SpikeProp and improves performance. Without approximation, Mostafa [108] used IF neurons to obtain an analytical expression relating input and output spike times. Comsa et al. [114,156] used SRM with biologically realistic alpha function for membrane dynamics and derived exact gradients with respect to input spike times and synaptic weights. Kheradpisheh and Masquelier [82] found that the first spike time of an IF neuron can approximate ReLU activation, they thus derived a new learning algorithm for multilayer SNNs with TTFS coding and achieved an accuracy of 97.4% on MNIST, which is comparable to [108,114].

Recent works extend the timing gradient approach to train deeper SNNs. To overcome gradient explosion and dead neuron problem encountered in timing gradient-based method, Zhang et al. [157] proposed a rectified linear postsynaptic potential function (ReL-PSP) based neuron model where membrane potential increases linearly prior to postsynaptic spike time. They trained convolutional SNNs, yielding 99.4% and 90.1% accuracy on MNIST and Fashion-MNITST, respectively. Zhou et al. [118] applied the method proposed by [108] to spiking VGG and GoogleNet with deep ANN training techniques such as batch normalization (BN), achieving an accuracy of 68.6% on ImageNet.

Activation gradient Activation gradient approaches smooth the hard non-linearity due to the none-or-one response of spiking neurons via modification of their activation function in forward or backward path.

The hard thresholds of spiking neurons can be placed with soft ones in forward propagation of SNNs. Huh and Sejnowski [109] introduced active zone where the synaptic current is activated gradually. They trained recurrent SNNs with backpropagation through time (BPTT) to perform dynamic tasks such as predictive coding. Liu et al. [158] found that all PCNN models can't explain the phenomenon that biological neurons stimulated by periodic signals exhibit chaotic behavior. They thus proposed a continuous-coupled neural network (CCNN) model to solve the problem. CCNN is a mean-field model where the step function used in PCNN is replaced by the sigmoid function which facilitates standard BP for training. CCNN also achieves better results in image segmentation tasks than state-of-the-art visual cortex models. These methods modify the binary output of spiking neurons, which is less friendly for low-power hardware implementation.

An alternative approach is to use surrogate gradients for backpropagation. SuperSpike [29] and STBP [30] were the early works applying gradient gradients to train multilayer SNNs. STBP uses the explicitly iterative LIF model. An SNN with iterative LIF can be unrolled in time as an RNN, which facilitates utilizing BPTT for the wight update. STBP updates the synaptic weights as follows

$$\Delta w_{ij}^l = -\eta \sum_t \left(\frac{\partial L}{\partial S_j^{t,l}} \frac{\partial S_j^{t,l}}{\partial V_j^{t,l}} + \frac{\partial L}{\partial V_j^{t+1,l}} \frac{\partial V_j^{t+1,l}}{\partial V_j^{t,l}} \right) \frac{\partial V_j^{t,l}}{\partial I_j^{t,l}} \frac{\partial I_j^{t,l}}{\partial w_{ij}^l} \tag{17}$$

where η is the learning rate. L is the loss function. $S_j^{t,l}$, $V_j^{t,l}$, and $I_j^{t,l}$ denote the output spike, the membrane potential, and the input current of neuron j at time t in layer l, respectively. w_{ij}^l is the synaptic weight from neuron i in $l-1$ layer to neuron j in l layer. In Eq. 17, the derivative of the spike function $\partial S_j^{t,l}/\partial V_j^{t,l}$ is not well defined. To enable gradient descent, a surrogate derivative is used for approximation. A common family of the surrogate gradient is the rectangular function [30] given by

$$h(V) = \frac{1}{a}\text{sign}\left(|V - V_{th}| < \frac{a}{2}\right) \tag{18}$$

where a is a hyper-parameter determining the shape for gradient estimation.

Shrestha and Orchard [159] considered the temporal dependence between spikes and interpreted the surrogate gradient as the probability density function. Wu et al. [85] extended STBP to deep convolutional SNNs with neuron normalization(NeuNorm). Zheng et al. [160] proposed threshold-dependent batch normalization (tdBN) for alleviating the gradient vanishing or explosion problem and maintaining the firing rate. Combined with STBP, tdBN realizes direct training of spiking ResNet-50 on ImageNet. As a time-dependent variant of tdBN, temporal effective batch normalization (TEBN) [161] can regularize the temporal distribution. To overcome the degradation problem in deep SNNs, Fang et al. [19] proposed SEW ResNet which can realize identity mapping. Feng et al. [162] proposed a multi-level firing (MLF) unit to combat the gradient vanishing problem. Hu et al. [20] scaled SNNs up to 104 layers on ImageNet and 482 layers on CIFAR-10, achieving 76.02% and 91.9% accuracy, respectively. The above studies have fully demonstrated the scalability of the surrogate gradient method.

The shape and smoothness of surrogate gradients have an impact on performance. To avoid the heuristic choice of surrogate gradients, Li et al. [163] proposed differentiable spike (Dspike) that can find the optimal shape and smoothness for surrogate gradients. Deng et al. [125] proposed temporal efficient training (TET) to solve the SNN generalization problem which results from incorrect surrogate gradients, obtaining a remarkable accuracy of 83% on CIFAR10-DVS.

The neuronal heterogeneity is a critical property of biological neural networks which can be incorporated into SNNs. Fang et al. [84] proposed a parametric LIF (PLIF) model which has learnable time constants. Wang et al. [164] proposed a learnable threshold scheme during training. Yao et al. [165] proposed a unified gated LIF (GLIF) neuron, wherein bio-features in different neural behaviors are fused by the gating factor. These methods can improve the learning of SNNs, thus obtaining better accuracy with fewer time steps.

Combined with BPTT, the surrogate gradient approach can achieve spatiotemporal credit assignment in very deep SNNs [20]. However, the large computational graph from the unrolled networks requires tremendous hardware resources during training. Local learning scheme provides an alternative to assign spatiotemporal credit in SNNs. Deep continuous local learning (DECOLLE) [166] uses layer-wise local readouts with random and fixed weights. Ma et al. [167] proposed a local learning scheme, wherein each layer is trained with a local auxiliary classifier. Inspired by the teacher-student learning approach, Yang et al. [168] proposed local tandem learning (LTL) to transfer the feature representation of a teacher ANN to a student SNN through layer-wise loss function. These local learning rules provide a competitive performance while consuming less computational costs than BPTT.

4.4.2. ANN pre-training

Leveraging the fact that the firing rate of IF models can approximate the activation value of ReLU functions, ANN pre-training approaches first train an ANN with constraints, then convert it into an SNN with or without post-conversion techniques fine-tuning the parameters.

The first ANN pre-training method was proposed by Cao et al. [27] which imposes constraints on CNN such as using ReLU activation function and removing biases to avoid negative outputs. Diehl et al. [102] suggested using weight normalization or threshold balancing to reduce conversion error, which is exactly equivalent mathematically [113]. Rueckauer et al. [105,83] proposed spike subtraction, also called soft reset, to alleviate information loss caused by resetting and implemented spiking equivalents of common operations such as BN which allow conversion of deeper CNNs including VGG-16 and GoogLeNet Inception-V3. Sengupta et al. [113] argued that removing the constraints in ANN training [83] suffers significant accuracy loss in the conversion process and proposed spike-norm to extend converted SNNs to residual architectures. RMP-SNN [115] used soft reset and a threshold balancing method that alleviates the firing rate vanishing problem. Hu et al. [121] proposed a compensation mechanism to reduce discretization errors and firstly built a converted SNN with more than 100 layers. Ding et al. [122] placed ReLU with rate norm layer (RLN) for ANN training, enabling direct conversion without setting thresholds manually.

In recent works, most studies focus on a theoretical analysis of conversion errors which facilitate methods reducing them and thus inference latency of the converted SNNs. Deng and Gu [119] added a threshold and shift in ReLU activation function to reduce conversion error. Li et al. [120] proposed SNN calibration which calibrates the parameters in converted SNN to match the activations in ANN. Bu et al. [124] used optimal initialization of membrane potentials to implement expected error-free conversion. Bu et al. [31] went deeper into error analysis and proposed a quantization clip-floor-shift activation function to replace the ReLU, achieving ultra-low latency (4 time steps) of converted SNNs.

There are also some interesting works that explore post-conversion fine-tuning of converted SNNs to reduce latency and increase accuracy. Rathi et al. [116] proposed a spiking-timing dependent surrogate function to train a converted SNN which converges within a few epochs and requires fewer time steps. Rathi and Roy [169] jointly optimized membrane leak and firing threshold along with synaptic weights to reduce latency and increases activation sparsity. Wu et al. [21] proposed a progressive tandem learning (PTL) framework that compensates the conversion errors layer-wise by tandem learning (TL) [117] with an adaptive training scheduler.

The ANN pre-training approach leverages the superior performance of ANNs while avoiding tremendous hardware overhead for direct training of SNNs, and thus quicker implementation on low-power neuromorphic hardware. However, this approach only works for static datasets so far and can't exploit the temporal dynamics of SNNs which makes them the third generation of neural networks [170].

4.4.3. ANN-coupled training

Unlike copying the weights from a well-trained ANN in the ANN pre-training approach, ANN-couple training consists of an SNN and an ANN with shared weights. The SNN feeds forward spike trains, while the ANN back propagates errors to update the shared weights. Wu et al. [117] first proposed this idea termed tandem learning in which the networks are coupled layer-wise. They demonstrated its effectiveness on both static and neuromorphic datasets. Kheradpisheh et al. [171] used a proxy ANN to backpropagate the errors of an SNN on the basis of rate-coded LF neurons approximating ReLU. They outperformed [117] on the CIFAR-10 dataset with an accuracy of 93.11%.

4.5. Performance comparison

So far, the majority of works on SNNs have used image classification datasets as benchmarks. To shed more light on the performance of SNN learning rules, we summarize the state-of-the-art results of existing methods tested on two types of datasets, i.e. static and neuromorphic datasets, which are presented in Table 4 and 5, respectively.

The static datasets in Table 4 include MNIST, CIFAR-10/100, and ImageNet. Due to the simplicity of MNIST, the recent accuracies reported on MNIST are pretty high ($> 99\%$). Therefore, CIFAR-10/100 and ImageNet have become more popular to evaluate deep SNNs. Recent results reported on these datasets are dominated by ANN pre-training [31,120] and activation gradient approach [20,125]. Though ANN pre-training methods are better than activation gradient-based methods in terms of accuracy, ANN pre-training methods need more time steps for inference.

Concerning the neuromorphic datasets, we include N-MNIST, CIFAR10-DVS, and DVS128 Gesture in Table 5. Similar to MNIST, the accuracies reported on N-MINST are above 99% in recent publications. The CIFAR10-DVS is a challenging dataset. TEBN [161] reported the best accuracy of 84.90% on CIFAR10-DVS with neuromorphic data augmentation [172]. Notably, almost all methods are based on activation gradient since they can exploit the temporal dynamics of the neuromorphic datasets.

5. Applications of SNNs

SNNs have potential advantages in representing and processing spatiotemporal patterns due to their inherent temporal dynamics. Thus, a vast body of studies explored the applications of SNNs in spatiotemporal tasks, such as event-based vision and audio signal processing. In this section, we introduce the applications of SNNs in these two fields.

Compared to conventional frame cameras, brain-inspired event cameras have several advantages, such as low power and high temporal resolution [9]. The SNNs are well-suited for processing the spatiotemporal data generated by event cameras due to their temporal dynamics. Orchard et al. [173] proposed an SNN for visual motion estimation, in which LIF neurons with synaptic delays implement motion-sensitive receptive fields. Haessig et al. [174] proposed an SNN-based direction sensitive (DS) unit for optical flow estimation. Paredes-Valles et al. [10] incorporated STDP learning to a hierarchical SNN which exhibits motion selectivity after training. Except for the above low-level vision applications, SNNs can also perform high-level vision tasks, such as recognition. The HMAX-inspired HFirst model [173] utilizes the spike timing provided by event cameras for the character recognition task. Xiao et al. [175] improved HFirst by integrating a tempotron classifier. Recent works [159,84,30,117] apply spike-based backpropagation to training deep SNNs, obtaining high accuracy on complex neuromorphic datasets created by event sensors [91,92].

Table 4
State-of-the-art results on the neuromorphic datasets, including MNIST, CIFAR-10/100, and ImageNet. – denotes the number of time steps is not reported in the paper or is not applicable to the method. T denotes the number of time steps. † denotes local learning rules.

Dataset	Reference	Year	Learning Rule	Accuracy	T
MNIST	Diehl's method [50]	2015	STDP-based	95.00	–
	Lee's method [107]	2016	Voltage gradient	99.31	–
	SDNN [28]	2018	STDP-based	98.40	–
	STBP [30]	2018	Activation gradient	99.42	–
	Zhou's method [118]	2021	Timing gradient	99.33	–
	RNL [122]	2021	ANN pre-training	99.46	–
	PLIF [84]	2021	Activation gradient	99.72	8
	STDBP [157]	2022	Timing gradient	99.40	–
CIFAR-10	NeuNorm [85]	2019	Activation gradient	90.53	12
	RMP-SNN [115]	2020	ANN pre-training	93.39	512
	TL [117]	2021	ANN-coupled training	90.98	8
	tdBN [160]	2021	Activation gradient	93.16	6
	Zhou's method [118]	2021	Timing gradient	92.68	–
	PLIF [84]	2021	Activation gradient	93.50	8
	Dspike [89]	2021	Activation gradient	94.25	6
	Calibration [120]	2021	ANN pre-taining	95.79	256
	Proxy [171]	2022	ANN-coupled training	93.11	60
	TET [125]	2022	Activation gradient	94.50	6
	QCFS [31]	2022	ANN pre-taining	96.08	32
	MLF [162]	2022	Activation gradient	94.25	4
	TEBN [161]	2022	Activation gradient	95.60	6
	LTL [168]	2022	Activation gradient†	95.28	32
CIFAR-100	RMP-SNN [115]	2020	ANN pre-training	70.58	1024
	DIET-SNN [169]	2021	ANN pre-training	69.67	5
	S-ResNet [121]	2021	ANN pre-training	70.62	350
	RNL [122]	2021	ANN pre-training	75.10	–
	Calibration [120]	2021	ANN pre-training	77.30	128
	TET [125]	2022	Activation gradient	74.72	6
	QCFS [31]	2022	ANN pre-training	79.62	32
	TEBN [161]	2022	Activation gradient	78.76	6
	LTL [168]	2022	Activation gradient†	76.08	32
ImageNet	RMP-SNN [115]	2020	ANN pre-training	73.09	2048
	S-ResNet [121]	2021	ANN pre-traning	73.77	350
	tdBN [160]	2021	Activation gradient	67.05	6
	Zhou's method [118]	2021	Timing gradient	68.80	–
	SEW-ResNet [19]	2021	Activation gradient	69.26	4
	MS-ResNet [20]	2021	Activation gradient	76.02	5
	Calibration [120]	2021	ANN pre-training	77.50	256
	TET [125]	2022	Activation activation	68.00	4
	QCFS [31]	2022	ANN pre-training	74.22	256
	TEBN [161]	2022	Activation gradient	68.28	4
	GLIF [165]	2022	Activation gradient	69.09	6

Table 5
State-of-the-art results on the neuromorphic datasets, including N-MNIST, CIFAR10-DVS, and DVS128 Gesture. – denotes the number of time steps is not reported in the paper or is not applicable to the method. T denotes the number of time steps. † denotes neuromorphic data augmentation.

Dataset	Reference	Year	Learning Rule	Accuracy	T
N-MNIST	Lee's Method [107]	2018	Voltage gradient	98.74	–
	STBP [30]	2018	Activation gradient	98.78	–
	SLAYER [159]	2018	Activation gradient	99.20	–
	NeuNorm [85]	2019	Activation gradient	99.53	–
	TL [117]	2021	ANN-coupled traning	99.31	–
	PLIF [84]	2021	Activation gradient	99.61	10
	LTMD [164]	2022	Activation gradient	99.65	15
CIFAR10-DVS	NeuNorm [85]	2019	Activation gradient	60.50	–
	TL [117]	2021	ANN-coupled traning	65.59	–
	tdBN [160]	2021	Activation gradient	67.80	10
	PLIF [84]	2021	Activation gradient	74.80	20
	Dspike [89]	2021	Activation gradient	75.40	10
	SEW-ResNet [19]	2021	Activation gradient	74.40	16
	TET [125]	2022	Activation gradient	77.33/83.17†	10
	TEBN [161]	2022	Activation gradient	84.90†	10
	GLIF [165]	2022	Activation gradient	78.10	16
DVS128 Gesture	SLAYER [159]	2018	Activation gradient	93.64	–
	tdBN [160]	2021	Activation gradient	96.87	40
	PLIF [84]	2021	Activation gradient	97.57	20
	SEW-ResNet [19]	2021	Activation gradient	97.92	16
	MLF[162]	2022	Activation gradient	97.29	40

SNN-based acoustic models have shown great potential for energy-efficient and high performance auditory information processing tasks, such as automatic sound classification (ASC) [176–178], automatic speech recognition (ASR) [179], and sound source localization (SSL) [180]. Several works [176–178] have successfully applied tempotron-based learning for ASC tasks. In these works, conventional feature extraction methods, such as mel-frequency cepstral coefficient (MFCC) and self-organizing map (SOM), are used for generating spike patterns, which are then classified by SNNs trained with the tempotron-based learning rule. Tavanaei and Maida [181,182] applied STDP to features extraction from raw speech signals. The extracted spike features are then post-processed into real-valued feature vectors and classified by a traditional classifier, such as SVM. The biological plausibility of these models is further improved by [183], wherein a fully SNN-based ASC framework is presented by combining competitive STDP learning and tempotron-based classification. Recent studies apply deep SNNs to auditory systems. Pan et al. [180] utilized surrogate gradient learning for SSL tasks. Deep SNNs trained with tandem learning rule have been explored for speech separation [21] and large vocabulary ASR [179].

6. Conclusion and future research directions

We presented a comprehensive survey of SNN learning rules, in which we reviewed the most representative learning rules in SNNs and provided discussions on their characteristics, advantages, limitations, and performance on several popular datasets. Besides, we introduced practical applications of SNNs in event-based vision and audio signal processing. Here, we further discuss a few challenges and promising research directions which may foster real-world applications in the field.

1. Learn lessons from deep learning. As we can see from the paper, the performance of SNNs has improved a lot by utilizing deep learning techniques, such as BPTT and BN. Even though there is still a gap between ANNs and SNNs in terms of accuracy, deep learning techniques have great potential to may further improve the performance of SNNs. On the one hand, the unique characteristics of SNNs, which may increase their performance, can be explored via deep learning techniques. For example, recent works show that neural architecture search (NAS) [184] can be exploited for finding better SNN architectures [185,186]. On the other hand, the temporal dimension and spike representation of SNNs should be considered when applying training techniques in deep learning to SNNs. For example, training deep SNNs using BPTT is constrained to working on dense tensors, limiting training speed and efficiency [187]. Standard BN in SNNs does not show scalability to large-scale datasets [188].
2. Draw inspirations from brain science. Biological observations and mechanisms of the human brain are natural references to creating intelligence. For example, burst-dependent plasticity can solve the credit assignment problem in hierarchical networks [142]. Dendritic spines [189] can facilitate weight optimization for pruning [190]. LIF neurons with lateral interactions [17] have shown improved robustness to noisy inputs. However, current SNNs mainly utilize single neuron dynamics such as LIF, while network-level dynamics, for instance, coherent oscillations [126] and chaos [191], are less explored in SNNs [158]. Although PCNNs are more biologically plausible [14] and the coupling mechanism has the potential for robust pattern recognition [17,158], the learning rules for them, as this survey revealed, have not been well developed. We expect that exploring learning rules in PCNNs may help us understand how the brain works and further improve the robustness of existing methods.
3. Algorithm-hardware co-design for energy-efficient neuromorphic systems. In this methodology, design goals are achieved by exploiting the synergism between algorithms and hardware platforms. The SNNs are spatiotemporal networks that are not well-suited for simulation on conventional hardware. Thus, it is necessary to explore the efficient implementation of the SNNs and their learning rules. On the one hand, spike-based computation in SNNs can be utilized for low-power neuromorphic chips [192,193]. On the other hand, novel devices may leverage more brain-like systems due to their unique merits. For instance, the memristor has nonlinearity, non-volatility, and is compatible with the CMOS technology [194]. It can emulate the complex nonlinear dynamics of biological neurons and synapses [135,195,196,134] and learning in SNNs [197,198]. We expect this paradigm to gain increased popularity in the near future.

Data availability

No data was used for the research described in the article.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study is supported by the Natural Science Foundation of Gansu Province (No. 21JR7RA510, No. 21JR7RA345) and the National Natural Science Foundation of China (No. 62276238, No. 62061023, No. 62272229).

References

[1] D. Cox, T. Dean, Neural Networks and Neuroscience-Inspired Computer Vision, Curr. Biol. 24 (18) (2014) R921–R929, ISSN 09609822..

[2] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386–408, ISSN 1939-1471, 0033-295X.

[3] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, ISSN 00189219..

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.U. Kaiser, I. Polosukhin, Attention is All you Need, Advances in Neural Information Processing Systems, vol. 30, Curran Associates Inc, 2017.

[5] Y. Bengio, Y. Lecun, G. Hinton, Deep Learning for AI, Commun. ACM 64 (7) (2021) 58–65, ISSN 0001–0782..

[6] E. Strubell, A. Ganesh, A. McCallum, Energy and Policy Considerations for Deep Learning in NLP, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 3645–3650, 2019.

[7] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, A survey on adversarial attacks and defences, CAAI Trans. Intell. Technol. 6 (1) (2021) 25–45.

[8] L. Zhu, S. Dong, J. Li, T. Huang, Y. Tian, Ultra-high Temporal Resolution Visual Reconstruction from a Fovea-like Spike Camera via Spiking Neuron Model, IEEE Trans. Pattern Anal. Mach. Intell. (2022) 1–1ISSN 0162–8828, 2160–9292, 1939–3539.

[9] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A.J. Davison, J. Conradt, K. Daniilidis, D. Scaramuzza, Event-Based Vision: A Survey, IEEE Trans. Pattern Anal. Mach. Intell. 44 (1) (2022) 154–180, ISSN 0162–8828, 2160–9292, 1939–3539.

[10] F. Paredes-Valles, K.Y.W. Scheper, G.C.H.E. de Croon, Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation: From Events to Global Motion Perception, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 2051–2064, ISSN 0162–8828, 2160–9292, 1939–3539.

[11] A.L. Hodgkin, A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. 117 (4) (1952) 500–544, ISSN 0022-3751, 1469-7793.

[12] E. Izhikevich, Simple model of spiking neurons, IEEE Trans. Neural Networks 14 (6) (2003) 1569–1572, ISSN 1045–9227..

[13] R. VanRullen, R. Guyonneau, S.J. Thorpe, Spike times make sense, Trends Neurosci. 28 (1) (2005) 1–4, ISSN 0166–2236..

[14] R. Eckhorn, H.J. Reitboeck, M. Arndt, P. Dicke, Feature Linking, Feature Linking via Synchronization among Distributed Assemblies: Simulations of Results from Cat Visual Cortex, Neural Computation 2 (3) (1990), 293–307, ISSN 0899-7667, 1530-888X.

[15] N.K. Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, Neural Networks 52 (2014) 62–76, ISSN 0893–6080..

[16] Z. Yu, J.K. Liu, S. Jia, Y. Zhang, Y. Zheng, Y. Tian, T. Huang, Toward the Next Generation of Retinal Neuroprosthesis: Visual Computation with Spikes, Engineering 6 (4) (2020) 449–461, ISSN 2095–8099..

[17] X. Cheng, Y. Hao, J. Xu, B. Xu, LISNN: Improving Spiking Neural Networks with Lateral Interactions for Robust Object Recognition, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Yokohama, Japan, 1519–1525, ISBN 978-0-9992411-6-5, 2020.

[18] X. She, Y. Long, D. Kim, S. Mukhopadhyay, ScieNet: Deep learning with spike-assisted contextual information extraction, Pattern Recogn. 118 (2021) 108002, ISSN 00313203.

[19] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, Y. Tian, Deep Residual Learning in Spiking Neural Networks, Advances in Neural Information Processing Systems, vol. 34, Curran Associates Inc, 2021, pp. 21056–21069.

[20] Y. Hu, Y. Wu, L. Deng, G. Li, Advancing Residual Learning towards Powerful Deep Spiking Neural Networks, arXiv:2112.08954 [cs], 2021a.

[21] J. Wu, C. Xu, X. Han, D. Zhou, M. Zhang, H. Li, K.C. Tan, Progressive Tandem Learning for Pattern Recognition with Deep Spiking Neural Networks, IEEE Trans. Pattern Anal. Mach. Intell. (2021a) 1–1ISSN 0162-8828, 2160-9292, 1939-3539.

[22] S.M. Bohte, J.N. Kok, H. La Poutre, Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing 48 (1–4) (2002) 17–37, publisher: Elsevier.

[23] R. Gütig, H. Sompolinsky, The tempotron: a neuron that learns spike timing–based decisions, Nature Neurosci. 9(3) (2006) 420–428, ISSN 1097-6256, 1546-1726.

[24] F. Ponulak, A. Kasiński, Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification, and Spike Shifting, Neural Comput. 22 (2) (2010) 467–510, ISSN 0899-7667, 1530-888X.

[25] A. Mohemmed, S. Schliebs, S. Matsuda, N. Kasabov, Span: Spike pattern association neuron for learning spatio-temporal spike patterns, Int. J. Neural Syst. 22 (04) (2012) 1250012.

[26] T. Masquelier, S.J. Thorpe, Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity, PLoS Comput. Biol. 3 (2) (2007), e31, ISSN 1553–7358.

[27] Y. Cao, Y. Chen, D. Khosla, Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition, Int. J. Comput. Vision 113 (1) (2015) 54–66, ISSN 0920-5691, 1573-1405.

[28] S.R. Kheradpisheh, M. Ganjtabesh, S.J. Thorpe, T. Masquelier, STDP-based spiking deep convolutional neural networks for object recognition, Neural Networks 99 (2018) 56–67, ISSN 08936080..

[29] F. Zenke, S. Ganguli, SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks, Neural Comput. 30(6) (2018) 1514–1541, ISSN 0899-7667, 1530-888X.

[30] Y. Wu, L. Deng, G. Li, J. Zhu, L. Shi, Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks, Front. Neurosci. 12 (2018), 331, ISSN 1662–453X.

[31] T. Bu, W. Fang, J. Ding, P. DAI, Z. Yu, T. Huang, Optimal ANN-SNN Conversion for High-accuracy and Ultra-low-latency Spiking Neural Networks, in: International Conference on Learning Representations, 2022a.

[32] A. Tavanaei, M. Ghodrati, S.R. Kheradpisheh, T. Masquelier, A. Maida, Deep learning in spiking neural networks, Neural Networks 111 (2019) 47–63, ISSN 0893–6080..

[33] E.O. Neftci, H. Mostafa, F. Zenke, Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks, IEEE Signal Process. Mag. 36 (6) (2019) 51–63, ISSN 1053-5888, 1558-0792.

[34] H. Jang, O. Simeone, B. Gardner, A. Gruning, An Introduction to Probabilistic Spiking Neural Networks: Probabilistic Models, Learning Rules, and Applications, IEEE Signal Process. Mag. 36 (6) (2019) 64–77, ISSN 1053-5888, 1558-0792.

[35] K. Roy, A. Jaiswal, P. Panda, Towards spike-based machine intelligence with neuromorphic computing, Nature 575 (7784) (2019) 607–617, ISSN 0028-0836, 1476-4687.

[36] X. Wang, X. Lin, X. Dang, Supervised learning in spiking neural networks: A review of algorithms and evaluations, Neural Networks 125 (2020) 258–280, ISSN 0893–6080..

[37] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L.P. Maguire, T. McGinnity, A review of learning in biologically plausible spiking neural networks, Neural Networks 122 (2020) 253–272, ISSN 0893–6080..

[38] J.L. Lobo, J. Del Ser, A. Bifet, N. Kasabov, Spiking Neural Networks and online learning: An overview and perspectives, Neural Networks 121 (2020) 88–100, ISSN 0893–6080..

[39] A. Javanshir, T.T. Nguyen, M.A.P. Mahmud, A.Z. Kouzani, Advancements in Algorithms and Neuromorphic Hardware for Spiking Neural Networks, Neural Computation 34 (6) (2022) 1289–1328, ISSN 0899-7667, 1530-888X.

[40] D. Zhang, T. Zhang, S. Jia, Q. Wang, B. Xu, Recent Advances and New Frontiers in Spiking Neural Networks, arXiv:2204.07050 [cs], 2022a.

[41] S. Wang, T.H. Cheng, M.H. Lim, A hierarchical taxonomic survey of spiking neural networks, Memetic Comput. 14 (3) (2022) 335–354, ISSN 1865-9284, 1865-9292.

[42] J. Johnson, M. Padgett, PCNN models and applications, IEEE Trans. Neural Networks 10 (3) (1999) 480–498, ISSN 10459227..

[43] Z. Wang, Y. Ma, F. Cheng, L. Yang, Review of pulse-coupled neural networks, Image Vision Comput. 28 (1) (2010a) 5–13, ISSN 02628856.

[44] K. Zhan, J. Shi, H. Wang, Y. Xie, Q. Li, Computational Mechanisms of Pulse-Coupled Neural Networks: A Comprehensive Review, Arch. Comput. Methods Eng. 24 (3) (2017) 573–588, ISSN 1134-3060, 1886-1784.

[45] L. Abbott, Lapicque's introduction of the integrate-and-fire model neuron (1907), Brain Res. Bull. 50 (5–6) (1999) 303–304, ISSN 03619230..

[46] W. Gerstner, W.M. Kistler, Spiking Neuron Models: Single Neurons, Populations, Plasticity, 1 edn.,., Cambridge University Press, 2002, ISBN 978-0-521-81384-6 978-0-521-89079-3 978-0-511-81570-6, 2002..

[47] H. Fang, A. Shrestha, Z. Zhao, Q. Qiu, Exploiting Neuron and Synapse Filter Dynamics in Spatial Temporal Learning of Deep Spiking Neural Network, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Yokohama, Japan, 2799–2806, ISBN 978-0-9992411-6-5, 2020.

[48] D.J. Saunders, D. Patel, H. Hazan, H.T. Siegelmann, R. Kozma, Locally connected spiking neural networks for unsupervised feature learning, Neural Networks 119 (2019) 332–340, ISSN 08936080..

[49] W. Zhang, P. Li, Skip-Connected Self-Recurrent Spiking Neural Networks With Joint Intrinsic Parameter and Synaptic Weight Training, Neural Computat. 33 (7) (2021) 1886–1913, ISSN 0899-7667, 1530-888X.

[50] P.U. Diehl, M. Cook, Unsupervised learning of digit recognition using spike-timing-dependent plasticity, Front. Computat. Neurosci. 9, ISSN 1662–5188.

[51] K. Zhan, H. Zhang, Y. Ma, New spiking cortical model for invariant texture retrieval and image processing, IEEE Trans. Neural Networks 20 (12) (2009) 1980–1986.

[52] J. Lian, Z. Yang, J. Liu, W. Sun, L. Zheng, X. Du, Z. Yi, B. Shi, Y. Ma, An Overview of Image Segmentation Based on Pulse-Coupled Neural Network, Arch. Comput. Methods Eng. 28 (2) (2021) 387–403, ISSN 1134-3060, 1886-1784.

[53] K. Zhan, J. Shi, Q. Li, J. Teng, M. Wang, Image segmentation using fast linking SCM, in: 2015 International Joint Conference on Neural Networks (IJCNN), IEEE, Killarney, 1–8, ISBN 978-1-4799-1960-4, 2015.

[54] Z. Wang, Y. Ma, J. Gu, Multi-focus image fusion using PCNN, Pattern Recogn. 43 (6) (2010) 2003–2016, ISSN 00313203..

[55] W. Huang, Z. Jing, Multi-focus image fusion using pulse coupled neural network, Pattern Recogn. Lett. 28 (9) (2007) 1123–1132, ISSN 01678655..

[56] M. Li, W. Cai, Z. Tan, A region-based multi-sensor image fusion scheme using pulse-coupled neural network, Pattern Recogn. Lett. 27 (16) (2006) 1948–1956, ISSN 01678655..

[57] J. Lian, J. Liu, Z. Yang, Y. Qi, H. Zhang, M. Zhang, Y. Ma, A Pulse-Number-Adjustable MSPCNN and Its Image Enhancement Application, IEEE Access 9 (2021b) 161069–161086, ISSN 2169–3536.

[58] K. Zhan, J. Shi, J. Teng, Q. Li, M. Wang, F. Lu, Linking synaptic computation for image enhancement, Neurocomputing 238 (2017b) 1–12, ISSN 09252312.

[59] K. Zhan, J. Teng, J. Shi, Q. Li, M. Wang, Feature-Linking Model for Image Enhancement, Neural Computation 28 (6) (2016) 1072–1100, ISSN 0899-7667, 1530-888X.

[60] K. Zhan, J. Teng, Y. Ma, Spiking Cortical Model for Rotation and Scale Invariant Texture Retrieval., J. Inf. Hiding Multim, Signal Process. 4 (3) (2013) 155–165.

[61] H. Li, X. Jin, N. Yang, Z. Yang, The recognition of landed aircrafts based on PCNN model and affine moment invariants, Pattern Recogn. Lett. 51 (2015) 23–29, ISSN 01678655..

[62] K. Waldemark, T. Lindblad, V. Bečanović, J.L. Guillen, P.L. Klingner, Patterns from the sky, Pattern Recogn. Lett. 21 (3) (2000) 227–237, ISSN 01678655..

[63] X. Deng, Y. Ma, M. Dong, A new adaptive filtering method for removing salt and pepper noise based on multilayered PCNN, Pattern Recogn. Lett. 79 (2016) 8–17, ISSN 01678655..

[64] M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex, Nature Neurosci. 2 (11) (1999) 1019–1025, ISSN 1097-6256, 1546-1726.

[65] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust Object Recognition with Cortex-Like Mechanisms, IEEE Trans. Pattern Anal. Mach. Intell. 29 (3) (2007) 411–426, ISSN 0162-8828, 2160-9292.

[66] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015) 85–117, ISSN 08936080..

[67] W. Gerstner, W.M. Kistler, R. Naud, L. Paninski, Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition, 1 edn.,., Cambridge University Press, 2014, ISBN 978-1-107-06083-8 978-1-107-63519-7 978-1-107-44761-5.

[68] D. Hebb, The Organization of Behavior, 0 edn.,., Psychology Press, 2005, ISBN 978-1-4106-1240-3,.

[69] C.J. Shatz, The Developing Brain, Scientific American 267 (3) (1992) 60–67, ISSN 00368733, 19467087, publisher: Scientific American, a division of Nature America Inc.

[70] T.V.P. Bliss, T. Lømo, Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path, J. Physiol. 232 (2) (1973) 331–356, ISSN 00223751..

[71] G.-Q. Bi, M.-M. Poo, Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic

Cell Type, J. Neurosci. 18 (24) (1998) 10464–10472, ISSN 0270-6474, 1529-2401.
[72] W. Levy, O. Steward, Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus, Neuroscience 8 (4) (1983) 791–797, ISSN 03064522..
[73] S. Song, K.D. Miller, L.F. Abbott, Competitive Hebbian learning through spike-timing-dependent synaptic plasticity, Nature Neurosci. 3 (9) (2000) 919–926, ISSN 1097-6256, 1546-1726.
[74] R. Guyonneau, R. VanRullen, S.J. Thorpe, Neurons Tune to the Earliest Spikes Through STDP, Neural Comput. 17 (4) (2005) 859–879, ISSN 0899-7667, 1530-888X.
[75] P. Falez, P. Tirilly, I.M. Bilasco, P. Devienne, P. Boulet, Unsupervised visual feature learning with spike-timing-dependent plasticity: How far are we from traditional feature learning approaches?, Pattern Recogn 93 (2019) 418–429, ISSN 00313203..
[76] G.H. Seol, J. Ziburkus, S. Huang, L. Song, I.T. Kim, K. Takamiya, R.L. Huganir, H.-K. Lee, A. Kirkwood, Neuromodulators Control the Polarity of Spike-Timing-Dependent Synaptic Plasticity, Neuron 55 (6) (2007) 919–929, ISSN 0896–6273..
[77] F. Nadim, D. Bucher, Neuromodulation of neurons and synapses, Curr. Opin. Neurobiol. 29 (2014) 48–56, ISSN 0959–4388..
[78] N. Frémaux, W. Gerstner, Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules, Front. Neural Circuits 9, ISSN 1662–5110.
[79] Y. Hao, X. Huang, M. Dong, B. Xu, A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule, Neural Networks 121 (2020) 387–395, ISSN 08936080..
[80] M. Mozafari, S.R. Kheradpisheh, T. Masquelier, A. Nowzari-Dalini, M. Ganjtabesh, First-Spike-Based Visual Categorization Using Reward-Modulated STDP, IEEE Trans. Neural Networks Learn. Syst. 29 (12) (2018) 6178–6190, ISSN 2162–237X, 2162–2388.
[81] R.V. Rullen, S.J. Thorpe, Rate Coding Versus Temporal Order Coding: What the Retinal Ganglion Cells Tell the Visual Cortex, Neural Comput. 13 (6) (2001) 1255–1283, ISSN 0899–7667..
[82] S.R. Kheradpisheh, T. Masquelier, Temporal backpropagation for spiking neural networks with one spike per neuron, Int. J. Neural Syst. 30 (06) (2020), 2050027.
[83] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, S.-C. Liu, Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification, Front. Neurosci. 11 (2017) 682, ISSN 1662–453X.
[84] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, Y. Tian, Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, QC, Canada, 2021, pp. 2641–2651, ISBN 978-1-66542-812-5,.
[85] Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, L. Shi, Direct Training for Spiking Neural Networks: Faster, Larger, Better, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019a) 1311–1318, ISSN 2374–3468, 2159–5399.
[86] Y. Kim, H. Park, A. Moitra, A. Bhattacharjee, Y. Venkatesha, P. Panda, Rate Coding Or Direct Coding: Which One Is Better For Accurate, Robust, And Energy-Efficient Spiking Neural Networks?, in: ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 71–75, 2022a.
[87] A. Krizhevsky, V. Nair, G. Hinton, The CIFAR-10 dataset, online: http://www. cs. toronto. edu/kriz/cifar. html 55 (5).
[88] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, Li Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Miami, FL, 2009, pp. 248–255, ISBN 978-1-4244-3992-8,.
[89] L. Deng, Y. Wu, X. Hu, L. Liang, Y. Ding, G. Li, G. Zhao, P. Li, Y. Xie, Rethinking the performance comparison between SNNS and ANNS, Neural Networks 121 (2020) 294–307, ISSN 08936080..
[90] W. He, Y. Wu, L. Deng, G. Li, H. Wang, Y. Tian, W. Ding, W. Wang, Y. Xie, Comparing SNNs and RNNs on neuromorphic vision datasets: Similarities and differences, Neural Networks 132 (2020) 108–120, ISSN 08936080..
[91] G. Orchard, A. Jayawant, G.K. Cohen, N. Thakor, Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades, Front. Neurosci. 9, ISSN 1662–453X.
[92] H. Li, H. Liu, X. Ji, G. Li, L. Shi, CIFAR10-DVS: An Event-Stream Dataset for Object Classification, Front. Neurosci. 11 (2017) 309, ISSN 1662–453X.
[93] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, D. Modha, A Low Power, Fully Event-Based Gesture Recognition System, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, 2017, pp. 7388–7397, ISBN 978-1-5386-0457-1,.
[94] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, Y. Andreopoulos, Graph-Based Object Classification for Neuromorphic Vision Sensing, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Seoul, Korea (South), 491–501, ISBN 978-1-72814-803-8, 2019.
[95] M. Zhang, H. Qu, A. Belatreche, X. Xie, EMPD: An Efficient Membrane Potential Driven Supervised Learning Algorithm for Spiking Neurons, IEEE Trans. Cognit. Develop. Syst. 10 (2) (2018) 151–162, ISSN 2379-8920, 2379-8939.
[96] J.-P. Pfister, T. Toyoizumi, D. Barber, W. Gerstner, Optimal Spike-Timing-Dependent Plasticity for Precise Action Potential Firing in Supervised Learning, Neural Comput. 18 (6) (2006) 1318–1348, ISSN 0899-7667, 1530-888X.
[97] S. Ghosh-Dastidar, H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, Neural Networks 22 (10) (2009) 1419–1431, ISSN 08936080..
[98] R.V. Florian, The Chronotron: A Neuron That Learns to Fire Temporally Precise Spike Patterns, PLoS ONE 7 (8) (2012) e40233, ISSN 1932–6203.
[99] Q. Yu, H. Tang, K.C. Tan, H. Li, Precise-Spike-Driven Synaptic Plasticity: Learning Hetero-Association of Spatiotemporal Spike Patterns, PLoS ONE 8 (11) (2013a) e78318, ISSN 1932–6203.
[100] J. Wang, A. Belatreche, L. Maguire, T.M. McGinnity, An online supervised learning method for spiking neural networks with adaptive structure, Neurocomputing 144 (2014) 526–536, ISSN 09252312..
[101] B. Gardner, I. Sporea, A. Grüning, Learning Spatiotemporally Encoded Pattern Transformations in Structured Spiking Neural Networks, Neural Computation 27 (12) (2015) 2548–2586, ISSN 0899-7667, 1530-888X.
[102] P.U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, M. Pfeiffer, Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing, in: 2015 International Joint Conference on Neural Networks (IJCNN), IEEE, Killarney, Ireland, 1–8, ISBN 978-1-4799-1960-4, 2015.
[103] N. Anwani, B. Rajendran, NormAD - Normalized Approximate Descent based supervised learning rule for spiking neurons, in: 2015 International Joint Conference on Neural Networks (IJCNN), IEEE, Killarney, Ireland, 1–8, ISBN 978-1-4799-1960-4, 2015.
[104] R. Gütig, Spiking neurons can discover predictive features by aggregate-label learning, Science 351 (6277) (2016) aab4113, ISSN 0036-8075, 1095-9203.
[105] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, Theory and Tools for the Conversion of Analog to Spiking Convolutional Neural Networks, arXiv:1612.04052 [cs, stat], 2016.
[106] S.R. Kheradpisheh, M. Ganjtabesh, T. Masquelier, Bio-inspired unsupervised learning of visual features leads to robust invariant object recognition, Neurocomputing 205 (2016), 382–392, ISSN 09252312.
[107] J.H. Lee, T. Delbruck, M. Pfeiffer, Training Deep Spiking Neural Networks Using Backpropagation, Front. Neurosci.10, ISSN 1662–453X.
[108] H. Mostafa, Supervised Learning Based on Temporal Coding in Spiking Neural Networks, IEEE Trans. Neural Networks Learn. Syst. (2017) 1–9ISSN 2162–237X, 2162–2388.
[109] D. Huh, T.J. Sejnowski, Gradient Descent for Spiking Neural Networks, in: Advances in Neural Information Processing Systems 31 (NeurIPS 2018), vol. 31, Curran Associates Inc, Montréal, Canada, 2018.
[110] S.R. Kulkarni, B. Rajendran, Spiking neural networks for handwritten digit recognition–Supervised learning and network optimization, Neural Networks 103 (2018), 118–127, ISSN 08936080.
[111] Q. Yu, H. Li, K.C. Tan, Spike Timing or Rate? Neurons Learn to Make Decisions for Both Through Threshold-Driven Plasticity, IEEE Trans. Cybern. 49 (6) (2019) 2178–2189, ISSN 2168-2267, 2168-2275.
[112] C. Lee, G. Srinivasan, P. Panda, K. Roy, Deep spiking convolutional neural network trained with unsupervised spike-timing-dependent plasticity, IEEE Trans. Cognit. Develop. Syst. 11 (3) (2018) 384–394.
[113] A. Sengupta, Y. Ye, R. Wang, C. Liu, K. Roy, Going Deeper in Spiking Neural Networks: VGG and Residual Architectures, Front. Neurosci. 13 (2019) 95, ISSN 1662–453X.
[114] I.M. Comsa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, J. Alakuijala, Temporal coding in spiking neural networks with alpha synaptic function, in: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 8529–8533, 2020.
[115] B. Han, G. Srinivasan, K. Roy, RMP-SNN: Residual Membrane Potential Neuron for Enabling Deeper High-Accuracy and Low-Latency Spiking Neural Network, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 13555–13564.
[116] N. Rathi, G. Srinivasan, P. Panda, K. Roy, Enabling Deep Spiking Neural Networks with Hybrid Conversion and Spike Timing Dependent Backpropagation, in: International Conference on Learning Representations, 2020.
[117] J. Wu, Y. Chua, M. Zhang, G. Li, H. Li, K.C. Tan, A Tandem Learning Rule for Effective Training and Rapid Inference of Deep Spiking Neural Networks, IEEE Trans. Neural Networks Learn. Syst. (2021) 1–15.
[118] S. Zhou, X. Li, Y. Chen, S.T. Chandrasekaran, A. Sanyal, Temporal-Coded Deep Spiking Neural Network with Easy Training and Robust Performance, Proceedings of the AAAI Conference on Artificial Intelligence 35 (12) (2021) 11143–11151, section: AAAI Technical Track on Machine Learning V.
[119] S. Deng, S. Gu, Optimal Conversion of Conventional Artificial Neural Networks to Spiking Neural Networks, in: International Conference on Learning Representations, 2021.
[120] Y. Li, S. Deng, X. Dong, R. Gong, S. Gu, A Free Lunch From ANN: Towards Efficient, Accurate Spiking Neural Networks//////editorration, in: Proceedings of the 38th International Conference on Machine Learning, vol. 139 of Proceedings of Machine Learning Research, PMLR, 6316–6325, 2021a.
[121] Y. Hu, H. Tang, G. Pan, Spiking Deep Residual Networks, IEEE Trans. Neural Networks Learn. Syst. (2021b) 1–6ISSN 2162–237X, 2162–2388.
[122] J. Ding, Z. Yu, Y. Tian, T. Huang, Optimal ANN-SNN Conversion for Fast and Accurate Inference in Deep Spiking Neural Networks, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Montreal, Canada, 2328–2336, ISBN 978-0-9992411-9-6, 2021.

[123] T. Zhang, X. Cheng, S. Jia, M.-M. Poo, Y. Zeng, B. Xu, Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks, Sci. Adv. 7 (43) (2021) eabh0146, ISSN 2375–2548.

[124] T. Bu, J. Ding, Z. Yu, T. Huang, Optimized Potential Initialization for Low-Latency Spiking Neural Networks, Proceedings of the AAAI Conference on Artificial Intelligence 36 (1) (2022b) 11–20, ISSN 2374–3468, 2159–5399.

[125] S. Deng, Y. Li, S. Zhang, S. Gu, Temporal Efficient Training of Spiking Neural Network via Gradient Re-weighting, in: International Conference on Learning Representations, 2022.

[126] R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk, H.J. Reitboeck, Coherent oscillations: A mechanism of feature linking in the visual cortex?: Multiple electrode and correlation analyses in the cat, Biological Cybernetics 60 (2) (1988) 121–130, ISSN 0340-1200, 1432-0770.

[127] R.M. Fitzsimonds, H.-J. Song, M.-M. Poo, Propagation of activity-dependent synaptic depression in simple neural networks, Nature 388 (6641) (1997) 439–448.

[128] B. Widrow, M.E. Hoff, Adaptive switching circuits, Stanford Univ Ca Stanford Electronics Labs, Tech. Rep., 1960.

[129] A. Tavanaei, A. Maida, BP-STDP: Approximating backpropagation using spike timing dependent plasticity, Neurocomputing 330 (2019), 39–47, ISSN 09252312.

[130] M. Beyeler, N.D. Dutt, J.L. Krichmar, Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule, Neural Networks 48 (2013) 109–124, ISSN 08936080.

[131] B. Illing, W. Gerstner, J. Brea, Biologically plausible deep learning – But how far can we go with shallow networks?, Neural Networks 118 (2019), 90–101, ISSN 08936080

[132] D. Querlioz, O. Bichler, P. Dollfus, C. Gamrat, Immunity to Device Variations in a Spiking Neural Network With Memristive Nanodevices, IEEE Trans. Nanotechnol. 12 (3) (2013) 288–295, ISSN 1536–125X, 1941–0085.

[133] D. Querlioz, O. Bichler, C. Gamrat, Simulation of a memristor-based spiking neural network immune to device variations, in: The 2011 International Joint Conference on Neural Networks, IEEE, San Jose, CA, USA, 1775–1781, ISBN 978-1-4244-9635-8, 2011.

[134] X. Fang, D. Liu, S. Duan, L. Wang, Memristive LIF Spiking Neuron Model and Its Application in Morse Code, Front. Neurosci. 16 (2022a) 853010, ISSN 1662–453X.

[135] X. Fang, S. Duan, L. Wang, Memristive FHN spiking neuron model and brain-inspired threshold logic computing, Neurocomputing 517 (2023), 93–105, ISSN 0925-2312.

[136] D.J. Saunders, H.T. Siegelmann, R. Kozma, M. Ruszinkao, STDP Learning of Image Patches with Convolutional Spiking Neural Networks, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, Rio de Janeiro, 1–7, ISBN 978-1-5090-6014-6, 2018.

[137] Q. Xu, J. Peng, J. Shen, H. Tang, G. Pan, Deep CovDenseSNN: A hierarchical event-driven dynamic framework with spiking neurons in noisy environment, Neural Networks 121 (2020), 512–519, ISSN 08936080.

[138] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S.J. Thorpe, T. Masquelier, Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks, Pattern Recogn. 94 (2019) 87–95, ISSN 00313203.

[139] Y. Xu, X. Zeng, S. Zhong, A New Supervised Learning Algorithm for Spiking Neurons, Neural Computation 25 (6) (2013) 1472–1511, ISSN 0899-7667, 1530-888X.

[140] A. Taherkhani, A. Belatreche, Y. Li, L.P. Maguire, DL-ReSuMe: A Delay Learning-Based Remote Supervised Method for Spiking Neurons, IEEE Trans. Neural Networks Learn. Syst. 26 (12) (2015) 3137–3149, ISSN 2162–237X, 2162–2388.

[141] M. Zhang, J. Wu, A. Belatreche, Z. Pan, X. Xie, Y. Chua, G. Li, H. Qu, H. Li, Supervised learning in spiking neural networks with synaptic delay-weight plasticity, Neurocomputing 409 (2020), 103–118, ISSN 09252312.

[142] A. Payeur, J. Guerguiev, F. Zenke, B.A. Richards, R. Naud, Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits, Nature Neurosci. 24 (7) (2021) 1010–1019, ISSN 1097-6256, 1546-1726.

[143] X. Xie, S. Wen, Z. Yan, T. Huang, Y. Chen, Designing pulse-coupled neural networks with spike-synchronization-dependent plasticity rule: image segmentation and memristor circuit application, Neural Comput. Appl. 32 (17) (2020) 13441–13452, ISSN 0941-0643, 1433-3058.

[144] H. Mostafa, G. Cauwenberghs, A Learning Framework for Winner-Take-All Networks with Stochastic Synapses, Neural Comput. 30 (6) (2018) 1542–1572, ISSN 0899–7667.

[145] J. Brea, W. Senn, J.-P. Pfister, Matching Recall and Storage in Sequence Learning with Spiking Neural Networks, J. Neurosci. 33 (23) (2013) 9565–9575, ISSN 0270–6474.

[146] D. Jimenez Rezende, W. Gerstner, Stochastic variational learning in recurrent spiking networks, Front. Comput. Neurosci. 8, ISSN 1662–5188.

[147] Q. Yu, S. Li, H. Tang, L. Wang, J. Dang, K.C. Tan, Toward Efficient Processing and Learning With Spikes: New Approaches for Multispike Learning, IEEE Trans. Cybern. 52 (3) (2022) 1364–1376, ISSN 2168-2267, 2168-2275.

[148] Q. Yu, J. Gao, J. Wei, J. Li, K.C. Tan, T. Huang, Improving Multispike Learning With Plastic Synaptic Delays, IEEE Trans. Neural Networks Learn. Syst. (2022b) 1–12ISSN 2162–237X, 2162–2388.

[149] M. Zhang, J. Wu, Y. Chua, X. Luo, Z. Pan, D. Liu, H. Li, Mpd-al: an efficient membrane potential driven aggregate-label learning algorithm for spiking neurons, Proceedings of the AAAI conference on artificial intelligence 33 (2019) 1327–1334.

[150] Q. Yu, H. Tang, K.C. Tan, H. Li, Rapid feedforward computation by temporal encoding and learning with spiking neurons, IEEE Trans. Neural Networks Learn. Syst. 24 (10) (2013) 1539–1552.

[151] Q. Xu, Y. Qi, H. Yu, J. Shen, H. Tang, G. Pan, CSNN: An Augmented Spiking based Framework with Perceptron-Inception, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, 1646–1652, ISBN 978-0-9992411-2-7, 2018.

[152] Q. Yu, S. Song, C. Ma, J. Wei, S. Chen, K.C. Tan, Temporal Encoding and Multispike Learning Framework for Efficient Recognition of Visual Patterns, IEEE Trans. Neural Networks Learn. Syst. (2021) 1–13.

[153] P. Panda, K. Roy, Unsupervised regenerative learning of hierarchical features in Spiking Deep Networks for object recognition, in: 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 299–306.

[154] M. Zhang, H. Qu, A. Belatreche, Y. Chen, Z. Yi, A Highly Effective and Robust Membrane Potential-Driven Supervised Learning Method for Spiking Neurons, IEEE Trans. Neural Networks Learn. Syst. 30 (1) (2019b) 123–137, ISSN 2162–237X, 2162–2388.

[155] R.-M. Memmesheimer, R. Rubin, B. Ölveczky, H. Sompolinsky, Learning Precisely Timed Spikes, Neuron 82 (4) (2014) 925–938, ISSN 08966273.

[156] I.-M. Comsa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, J. Alakuijala, Temporal Coding in Spiking Neural Networks With Alpha Synaptic Function: Learning With Backpropagation, IEEE Trans. Neural Networks Learn. Syst. (2021) 1–14ISSN 2162–237X, 2162–2388.

[157] M. Zhang, J. Wang, J. Wu, A. Belatreche, B. Amornpaisannon, Z. Zhang, V.P.K. Miriyala, H. Qu, Y. Chua, T.E. Carlson, H. Li, Rectified Linear Postsynaptic Potential Function for Backpropagation in Deep Spiking Neural Networks, IEEE Trans. Neural Networks Learn. Syst. 33 (5) (2022b) 1947–1958, ISSN 2162–237X, 2162–2388.

[158] J. Liu, J. Lian, J.C. Sprott, Q. Liu, Y. Ma, The Butterfly Effect in Primary Visual Cortex, IEEE Trans. Comput. (2022) 1–1ISSN 0018–9340, 1557–9956, 2326–3814.

[159] S.B. Shrestha, G. Orchard, SLAYER: Spike Layer Error Reassignment in Time, in: Advances in Neural Information Processing Systems, vol. 31, Curran Associates Inc, 2018.

[160] H. Zheng, Y. Wu, L. Deng, Y. Hu, G. Li, Going Deeper With Directly-Trained Larger Spiking Neural Networks, Proceedings of the AAAI Conference on Artificial Intelligence 35 (12) (2021) 11062–11070, ISSN 2374-3468, 2159-5399

[161] C. Duan, J. Ding, S. Chen, Z. Yu, T. Huang, Temporal Effective Batch Normalization in Spiking Neural Networks, in, Adv. Neural Inform. Process. Syst. (2022).

[162] L. Feng, Q. Liu, H. Tang, D. Ma, G. Pan, Multi-Level Firing with Spiking DS-ResNet: Enabling Better and Deeper Directly-Trained Spiking Neural Networks, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria, 2471–2477, ISBN 978-1-956792-00-3, 2022.

[163] Y. Li, Y. Guo, S. Zhang, S. Deng, Y. Hai, S. Gu, Differentiable Spike: Rethinking Gradient-Descent for Training Spiking Neural Networks, in: Advances in Neural Information Processing Systems, vol. 34, Curran Associates Inc, 23426–23439, 2021b.

[164] S. Wang, T.H. Cheng, M.-H. Lim, LTMD: Learning Improvement of Spiking Neural Networks with Learnable Thresholding Neurons and Moderate Dropout, Advances in Neural Information Processing Systems, in, 2022.

[165] X. Yao, F. Li, Z. Mo, J. Cheng, GLIF: A Unified Gated Leaky Integrate-and-Fire Neuron for Spiking Neural Networks, in, Adv. Neural Inform. Process. Syst. (2022).

[166] J. Kaiser, H. Mostafa, E. Neftci, Synaptic Plasticity Dynamics for Deep Continuous Local Learning (DECOLLE), Front. Neurosci. 14 (2020) 424, ISSN 1662–453X.

[167] C. Ma, R. Yan, Z. Yu, Q. Yu, Deep Spike Learning With Local Classifiers, IEEE Trans. Cybern. (2022) 1–13ISSN 2168–2267, 2168–2275.

[168] Q. Yang, J. Wu, M. Zhang, Y. Chua, X. Wang, H. Li, Training Spiking Neural Networks with Local Tandem Learning, Advances in Neural Information Processing Systems, in, 2022.

[169] N. Rathi, K. Roy, DIET-SNN: A Low-Latency Spiking Neural Network With Direct Input Encoding and Leakage and Threshold Optimization, IEEE Trans. Neural Networks Learn. Syst. (2021) 1–9.

[170] W. Maass, Networks of spiking neurons: The third generation of neural network models, Neural Networks 10 (9) (1997) 1659–1671, ISSN 0893–6080.

[171] S.R. Kheradpisheh, M. Mirsadeghi, T. Masquelier, Spiking Neural Networks Trained via Proxy, IEEE Access 10 (2022) 70769–70778, ISSN 2169–3536.

[172] Y. Li, Y. Kim, H. Park, T. Geller, P. Panda, Neuromorphic Data Augmentation for Training Spiking Neural Networks, arXiv preprint arXiv:2203.06145.

[173] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, R. Benosman, HFirst: A Temporal Approach to Object Recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (10) (2015b) 2028–2040, ISSN 0162–8828, 2160–9292, 1939–3539.

[174] G. Haessig, A. Cassidy, R. Alvarez, R. Benosman, G. Orchard, Spiking Optical Flow for Event-Based Sensors Using IBM's TrueNorth Neurosynaptic System, IEEE Trans. Biomed. Circuits Syst. 12 (4) (2018) 860–870, ISSN 1932-4545, 1940-9990.

[175] R. Xiao, H. Tang, Y. Ma, R. Yan, G. Orchard, An Event-Driven Categorization Model for AER Image Sensors Using Multispike Encoding and Learning, IEEE

Trans. Neural Networks Learn. Syst. 31(9) (2020) 3649–3657, ISSN 2162–237X, 2162–2388.

[176] J. Wu, Y. Chua, M. Zhang, H. Li, K.C. Tan, A Spiking Neural Network Framework for Robust Sound Classification, Front. Neurosci. 12 (2018b) 836, ISSN 1662–453X.

[177] J. Wu, Y. Chua, H. Li, A Biologically Plausible Speech Recognition Framework Based on Spiking Neural Networks, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, Rio de Janeiro, 1–8, ISBN 978-1-5090-6014-6, 2018c.

[178] R. Xiao, R. Yan, H. Tang, K.C. Tan, A Spiking Neural Network Model for Sound Recognition, in: Cognitive Systems and Signal Processing, vol. 710, Springer Singapore, Singapore, 584–594, ISBN 978-981-10-5229-3 978-981-10-5230-9, series Title: Communications in Computer and Information Science, 2017.

[179] J. Wu, E. Yılmaz, M. Zhang, H. Li, K.C. Tan, Deep Spiking Neural Networks for Large Vocabulary Automatic Speech Recognition, Front. Neurosci. 14 (2020) 199, ISSN 1662–453X.

[180] Z. Pan, M. Zhang, J. Wu, J. Wang, H. Li, Multi-Tone Phase Coding of Interaural Time Difference for Sound Source Localization With Spiking Neural Networks, IEEE/ACM Trans. Audio, Speech, Language Process. 29 (2021) 2656–2670, ISSN 2329-9290, 2329-9304.

[181] A. Tavanaei, A.S. Maida, A spiking network that learns to extract spike signatures from speech signals, Neurocomputing 240 (2017a) 191–199, ISSN 09252312.

[182] A. Tavanaei, A. Maida, Bio-inspired Multi-layer Spiking Neural Network Extracts Discriminative Features from Speech Signals, in: Neural Information Processing, vol. 10639, Springer International Publishing, Cham, 899–908, ISBN 978-3-319-70135-6 978-3-319-70136-3, 2017b.

[183] J. Wu, M. Zhang, H. Li, Y. Chua, Competitive STDP-based Feature Representation Learning for Sound Event Classification, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, Budapest, Hungary, 1–8, ISBN 978-1-72811-985-4, 2019b.

[184] T. Elsken, J.H. Metzen, F. Hutter, Neural Architecture Search: A Survey, J. Mach. Learn. Res. 20 (55) (2019) 1–21.

[185] B. Na, J. Mok, S. Park, D. Lee, H. Choe, S. Yoon, AutoSNN: Towards Energy-Efficient Spiking Neural Networks, in: Proceedings of the 39th International Conference on Machine Learning, vol. 162 of Proceedings of Machine Learning Research, PMLR, 16253–16269, 2022.

[186] Y. Kim, Y. Li, H. Park, Y. Venkatesha, P. Panda, Neural Architecture Search for Spiking Neural Networks, arXiv:2201.10355 [cs, eess], 2022b.

[187] N. Perez-Nieves, D.F.M. Goodman, Sparse Spiking Gradient Descent, in, Advances in Neural Information Processing Systems, 2021.

[188] Y. Kim, P. Panda, Revisiting Batch Normalization for Training Low-Latency Deep Spiking Neural Networks From Scratch, Front. Neurosci. 15 (2021) 773954, ISSN 1662–453X.

[189] R. Yuste, Dendritic Spines and Distributed Circuits, Neuron 71 (5) (2011) 772–781, ISSN 08966273.

[190] Y. Chen, Z. Yu, W. Fang, Z. Ma, T. Huang, Y. Tian, State Transition of Dendritic Spines Improves Learning of Sparse Spiking Neural Networks, in: Proceedings of the 39th International Conference on Machine Learning, vol. 162 of Proceedings of Machine Learning Research, PMLR, 3701–3715, 2022.

[191] R. Siegel, Non-linear dynamical system theory and primary visual cortical processing, Physica D: Nonlinear Phenomena 42 (1–3) (1990) 385–395, ISSN 01672789.

[192] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, B.L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S.K. Esser, R. Appuswamy, B. Taba, A. Amir, M.D. Flickner, W.P. Risk, R. Manohar, D.S. Modha, A million spiking-neuron integrated circuit with a scalable communication network and interface, Science 345 (6197) (2014) 668–673.

[193] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S.H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, H. Wang, Loihi: A Neuromorphic Manycore Processor with On-Chip Learning, IEEE Micro 38 (1) (2018) 82–99.

[194] S. Duan, X. Hu, Z. Dong, L. Wang, P. Mazumder, Memristor-Based Cellular Nonlinear/Neural Network: Design, Analysis, and Applications, IEEE Trans. Neural Networks Learn. Syst. 26 (6) (2015) 1202–1213, ISSN 2162–237X, 2162–2388.

[195] X. Fang, S. Duan, L. Wang, Memristive Izhikevich Spiking Neuron Model and Its Application in Oscillatory Associative Memory, Front. Neurosci. 16 (2022b) 885322, ISSN 1662–453X.

[196] X. Fang, S. Duan, L. Wang, Memristive Hodgkin-Huxley Spiking Neuron Model for Reproducing Neuron Behaviors, Front. Neurosci. 15 (2021c) 730566, ISSN 1662–453X.

[197] J. Li, Z. Dong, L. Luo, S. Duan, L. Wang, A novel versatile window function for memristor model with application in spiking neural network, Neurocomputing 405 (2020), 239–246, ISSN 09252312.

[198] N. Zheng, P. Mazumder, Learning in Memristor Crossbar-Based Spiking Neural Networks Through Modulation of Weight-Dependent Spike-Timing-Dependent Plasticity, IEEE Trans. Nanotechnol. 17 (3) (2018) 520–532, ISSN 1536–125X, 1941–0085.

[199] M. Zhang et al., An Efficient Threshold-Driven Aggregate-Label Learning Algorithm for Multimodal Information Processing, IEEE Journal of Selected Topics in Signal Processing 14 (3) (2020) 592–602, https://doi.org/10.1109/JSTSP.2020.2983547.

Zexiang Yi received the B.Eng. degree from Xidian University, Xi'an, China. He is currently pursuing the master's degree at Lanzhou University, Lanzhou, China. His current research interests include spiking neural networks and neuromorphic computing.

Jing Lian received the B.S. degree in signal and information processing from Lanzhou Jiaotong University in 2010, and the Ph.D. degree from Lanzhou University, Lanzhou, China, in 2017. He is currently an Associate Professor in the School of Electronics and Information Engineering, Lanzhou Jiaotong University. His research interests include artificial neural networks, medical image processing, and cognitive visual processing.

Qidong Liu received the Ph.D. degree from the School of Information Science & Engineering, Lanzhou University in 2018. From 2019 to 2021, he worked as a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently an Associate Professor at the School of Computer and Artificial Intelligence, Zhengzhou University, China. His research interests include machine learning, reinforcement learning, graph neural network, etc.

Hegui Zhu received the B.S. and M.S. degrees from Northeastern University, Shenyang, China, in 2003 and 2006, respectively, and the Ph.D. degree from Jilin University, Changchun, China, in 2014. He is currently an Associate Professor with the Department of Math, College of Sciences, Northeastern University. His current research interests include chaotic systems, chaos-based applications, and multimedia information security.

Dong Liang received the B.S. degree in telecommunication engineering and the M.S. degree in circuits and systems from Lanzhou University, Lanzhou, China, in 2008 and 2011, respectively, and the Ph.D. degree from the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan, in 2015. He is currently an Associate Professor with the Pattern Recognition and Neural Computing Laboratory, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include computer vision and pattern recognition.

Jizhao Liu received his B.S, M.S., and Ph.D. at Lanzhou University, Lanzhou, China. He was a postdoctoral fellow at Sun Yat-sen University. He is an associate professor with the School of Information Science and Engineering, Lanzhou University, China. His research has been primarily in chaos theory, chaotic circuits, and related engineering applications. His current research interests include the nonlinear characteristic in biological neural networks, neural coding, brain-like and brain-inspired computing.