Recent Advances and New Frontiers in Spiking Neural Networks

Duzhen Zhang[1,2]*, Shuncheng Jia[1,2], Qingyu Wang[1,2]

[1]Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China.
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences

{zhangduzhen2019}@ia.ac.cn

Abstract

In recent years, spiking neural networks (SNNs) have received extensive attention in brain-inspired intelligence due to their rich spatially-temporal dynamics, various encoding methods, and event-driven characteristics that naturally fit the neuromorphic hardware. With the development of SNNs, brain-inspired intelligence, an emerging research field inspired by brain science achievements and aiming at artificial general intelligence, is becoming hot. This paper reviews recent advances and discusses new frontiers in SNNs from five major research topics, including essential elements (i.e., spiking neuron models, encoding methods, and topology structures), neuromorphic datasets, optimization algorithms, software, and hardware frameworks. We hope our survey can help researchers understand SNNs better and inspire new works to advance this field.

1 Introduction

Brain science (BS) and artificial intelligence (AI) research have ushered in rapid development in mutual promotion, and brain-inspired intelligence research with interdisciplinary characteristics has received more and more attention. Brain-inspired intelligence aims to obtain inspiration from BS research regarding structure, mechanism, or function to improve AI. It enables AI to integrate various cognitive abilities and gradually approach or even surpass human intelligence in many aspects. Spiking neural networks (SNNs) are at the core of brain-inspired intelligence research. By emphasizing the highly brain-inspired structural basis and optimization algorithms, SNNs try to accelerate the understanding of the nature of biological intelligence from the perspective of computing, thereby laying a theoretical foundation for the formation of a new generation of human-level AI models.

As the main driving force of the current AI development, artificial neural networks (ANNs) have experienced multiple generations of evolution. The first generation of ANNs, called perceptron, can simulate human perception [Rosenblatt, 1958]. The second generation is connectionism-based deep neural networks (DNNs) that emerged in the mid-1980s [Rumelhart *et al.*, 1986] and have led the development of AI for the past dozen years since 2006 [Hinton *et al.*, 2006]. However, DNNs that transmit information primarily by firing rate are biologically imprecise and lack dynamic mechanisms within neurons. SNNs are considered the third generation of ANNs due to their rich spatially-temporal neural dynamics, diverse encoding methods, and event-driven advantages [Maass, 1997].

The proposal of SNNs marks the gradual transition of ANNs from spatial encoding dominated by firing rate to spatially-temporal hybrid encoding dominated by precise spike firing and subthreshold dynamic membrane potential. The newly added temporal dimension makes it possible for more precise biological computing simulation, more stable and robust information representation, and more energy-efficient network computation.

In this survey, we comprehensively review the recent advances of SNNs and focus on five major research topics, which we define as:

- **Essential Elements.** The essential elements of SNNs include the neuron models as the basic processing unit, the encoding methods of the spike trains in neuron communication, and the topology structures of each basic layer at the network level (see Section 2).

- **Neuromorphic Datasets.** The development of datasets has played an essential role in promoting the progress of ANNs. Currently, datasets suitable for SNNs are composed of spatially-temporal event streams, such as N-MNIST [Orchard *et al.*, 2015] and DVS-CIFAR10 [Li *et al.*, 2017] (see Section 3).

- **Optimization Algorithms.** How to efficiently optimize SNNs has been the focus of research in recent years. The research on optimization algorithms can be divided into two main types. One type is designed to understand better the biological system, such as spike-timing-dependent plasticity (STDP) [Bi and Poo, 1998]. The other type is constructed to pursue superior computational performance, such as pseudo backpropagation (BP) [Zenke and Ganguli, 2018] (see Section 4).

- **Software Frameworks.** Software frameworks are able to support the construction and training of SNNs, such

---

as SpikingJelly [Fang *et al.*, 2020] and CogSNN [Zhang and Liu, 2022] (see Section 5).

- **Hardware Frameworks.** Due to the advantage of ultra-low energy consumption of SNNs in hardware circuits, neuromorphic chips that support SNNs hardware implementation have sprung up, such as IBM TrueNorth [Akopyan *et al.*, 2015] and Intel Loihi [Davies *et al.*, 2018] (see Section 6).

We also have broad discussions on new frontiers in each topic. Finally, we summarize the paper (see Section 7). The overall architecture of SNNs is shown in Figure 1.

We hope this survey will help researchers understand the latest progress, challenges, and frontiers in the SNNs field.

2 Essential Elements

Essential elements include the neuron models as the basic information processing unit, the encoding methods of the spike trains in neuron communication, and the topology structures of each basic layer at the network level, which constitute the SNNs together.

2.1 Neuron Models

Recent Advances

The typical structure of biological neurons mainly includes three parts: dendrite, soma, and axon [Zhang *et al.*, 2021d]. The function of dendrites is to collect input signals from other neurons and transmit them to the soma. The soma acts as a central processor, generating spikes when the afferent currents cause the neuron membrane potential to exceed a certain threshold (i.e., action potential). The spikes propagate along the axon without attenuation and transmit signals to the next neuron through the synapse at the axon's end. According to the dynamic characteristics of the neuronal potential, neurophysiologists have established many neuron models, represented by the Hodgkin-Huxley(H-H) [Hodgkin and Huxley, 1952], the leaky integrate-and-fire (LIF) [Dayan *et al.*, 2003], and the Izhikevich [Izhikevich *et al.*, 2004] model.

By studying the potential data of squid axons, Hodgkin and Huxley [1952] proposed a theoretical mathematical model of the mechanism of neuronal electrical activity, called the H-H model, formulated as:

$$\frac{dV}{dt} = -g_{N_a}(V - V_{N_a}) - g_K(V - V_K) + I, \quad (1)$$

where V denotes membrane potential, g_{N_a} and g_K denote the conductance densities of sodium and potassium ions, V_{N_a} and V_K denote the reversal potentials of sodium and potassium ion channels, and I denotes total membrane current density.

Since little is known about the mechanism of action potential generation in the early years, the process of action potential generation was simplified as follows: "When the membrane potential exceeds the threshold V_{th}, the neuron will fire a spike, and the membrane potential falls back to the resting value V_{rest}". The LIF model follows this principle and introduces a leak factor, allowing the membrane potential to shrink over time [Dayan *et al.*, 2003]. It describes the change

law of membrane potential below the threshold. The simplest and most common form of the LIF model is formulated as:

$$\tau_m \frac{dV}{dt} = V_{rest} - V + R_m I, \quad (2)$$

where τ_m denotes membrane time constant, V_{rest} denotes resting potential, and I and R_m denote the input current and the impedance of the cell membrane, respectively. The LIF model simplifies the action potential generation process but retains the three critical features of the biological neuron, i.e., membrane potential leakage, integration accumulation, and threshold firing. Later variants of the LIF model further describe the details of neuronal spiking activity, enhancing its biological credibility.

Izhikevich model uses a limited number of dimensionless parameter combinations (such as a and b.) to characterize multiple types of rich spike firing patterns and can display the firing patterns of almost all known neurons in the cerebral cortex through the choice of parameters [Izhikevich *et al.*, 2004]. It is an efficient method for constructing second-order neural dynamics equations, formulated as:

$$\frac{dV}{dt} = 0.04V^2 + 5V + 140 - u + I, \quad (3)$$

$$\frac{du}{dt} = a(bV - u), \quad (4)$$

where u is a membrane recovery variable used to describe the ionic current behavior, a and b are used to adjust the time scale of u and the sensitivity to the membrane potential V.

By contrast, we briefly introduce the basic neuron model in DNNs. It retains the multi-input and single-output information processing form of biological neurons but further simplifies its threshold characteristics and action potential mechanism, formulated as:

$$y_i^l = \sigma(\sum_{j=0}^{n^{l-1}} w_{ij}^{l-1} x_j^{l-1}), \quad (5)$$

where after the weighted summation of the output values x_j^{l-1} of the n^{l-1} neurons in the previous layer, the nonlinear activation function $\sigma(\cdot)$ calculates the output value y_i^l of the ith neuron in the lth layer.

Compared with SNNs, DNNs use high-precision continuous floating-point values instead of discrete spike trains for communication, abandoning operations in the temporal domain and retaining only the spatial domain structure of the layer-by-layer computation. Although SNNs have lower expression precision, they keep richer neuron dynamics and are closer to biological neurons. In addition to receiving input in the spatial domain, the current state is also naturally influenced by past historical states. Therefore, SNNs have more substantial spatially-temporal information processing potential and biological plausibility. Due to threshold characteristics, the spike signal (0 or 1) of SNNs is usually very sparse, and the calculation is driven by events (only executed when the spike arrives), showing ultra-low power consumption and computational cost. Moreover, SNNs exhibit stronger anti-noise robustness than vulnerable DNNs. Individual neurons
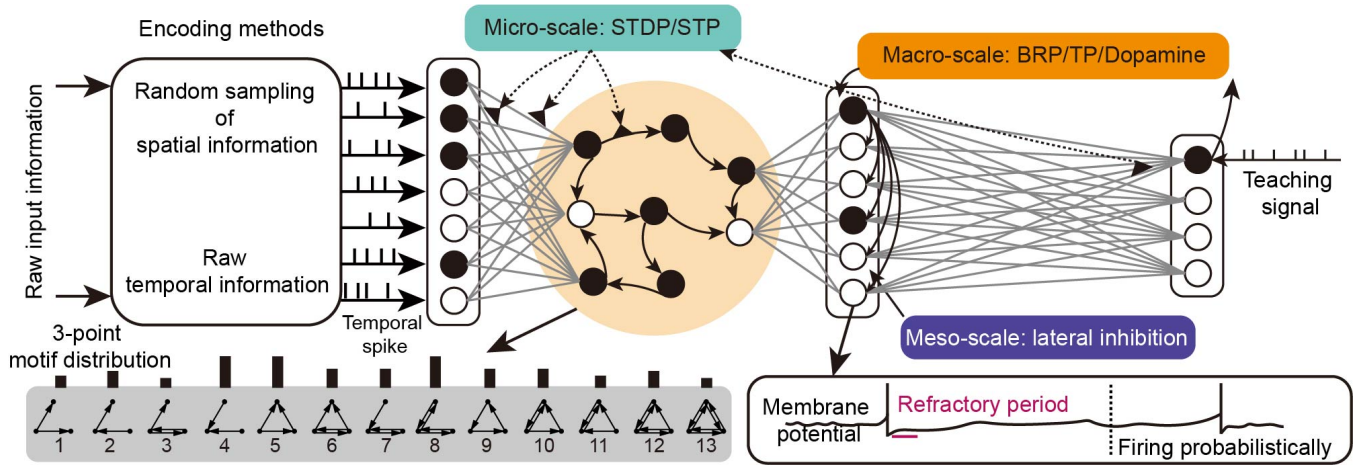
Figure 1: The overall architecture of SNNs, including encoding methods, motif topology, and multi-scale synaptic plasticity, etc.

operating in spikes act as microscopic bottlenecks that maintain low intermittent noise and do not transmit sub-threshold noise to their neighbors. In summary, the spiking communication and dynamic characteristics of SNNs' neurons constitute the most fundamental difference from DNNs, which endows them with the potential for spatially-temporal task processing, ultra-low power, and robust computing.

New Frontiers
There are multiple choices of different abstraction levels between bionic degree and computational complexity for modeling biological neurons. Complex models, e.g., the H-H model, that use multi-variable, multi-group differential equations for precise activity descriptions cannot be applied to large-scale neural networks. Therefore, it is indispensable to simplify the model to speed up the simulation process. At present, the widely adopted LIF model can guarantee a low computational cost, but it relatively lacks biological credibility. To ensure the ability to construct larger-scale neural networks, it is still an urgent problem to find a neuron model with both excellent learning ability and high biological credibility.

2.2 Encoding Methods

Recent Advances
At present, the common neural encoding methods mainly include rate, temporal, and population coding. Rate coding uses the firing rate of spike trains in a time window to encode information, where real input numbers are converted into spike trains with a frequency proportional to the input value [Adrian and Zotterman, 1926; Cheng *et al.*, 2020]. Temporal coding encodes information with the relative timing of individual spikes, where input values are usually converted into spike trains with the precise time, including time-to-first-spike coding [VanRullen *et al.*, 2005], rank order coding [Thorpe and Gautrais, 1998], etc. Besides that, population coding is special in integrating these two types. For example, each neuron in a population can generate spike trains with precise time and also contain a relation with other neurons for better information encoding at a global scale [Georgopoulos *et al.*, 1986; Zhang *et al.*, 2021a].

New Frontiers
Currently, the specific method of neural encoding has not yet been concluded. Populations of neurons with different encodings may coexist and cooperate, thus providing a sufficient perception of information. Neural encoding methods may behave differently in different brain regions. Compared to the limited case where current SNNs usually preset a single encoding method, more ideal and general SNNs should support hybrid applications of different encodings. They can utilize different encodings' advantages flexibly to optimize task performance, delay, and power consumption. Furthermore, many SNNs algorithms only pay attention to rate coding, ignoring the spike trains' temporal structure. It may cause that the advantages of SNNs in temporal information processing have not been well exploited. Therefore, the design of the algorithm suitable for temporal coding with high information density may be the new direction for future exploration.

2.3 Topology Structures

Recent Advances
Similar to DNNs, the basic topology used to construct SNNs includes a fully connected, recurrent, and convolutional layer. The corresponding neural networks are multi-layer perceptrons (MLPs), recurrent neural networks (RNNs), and convolutional neural networks (CNNs). MLPs and RNNs are mainly for one-dimensional feature processing, while CNNs are mainly for two-dimensional feature processing. RNNs can be regarded as MLPs that add recurrent connections, and they are especially good at processing temporal features.

New Frontiers
Compared to the structures in biological networks, the current topology of SNNs is relatively simplified. The structure of brain connections at different scales is very complex. Multipoint minimal motif network can be used as a primary network structure unit to analyze the functions of complex network systems [Sporns *et al.*, 2004]. As Figure 1 shows, taking the 3-point motif as an example, when the node types (such as different neuron types) are not considered, the combination of

different primitive motifs is limited to 13 categories. For networks that complete similar functions, the Motif distributions tend to have strong consistency and stability. For function-specific networks, the Motif distributions between them are pretty different [Sporns *et al.*, 2004]. Therefore, we can better understand their functions and connectivity patterns by analyzing the motif distributions in complex biological networks [Jia *et al.*, 2022]. Based on the parsed motif distributions, we can add constraints to the network structure design or search algorithms, thereby obtaining biologically plausible and interpretable new topology structures. Moreover, some recent methods propose an effective modeling framework to integrate SNNs with graph neural networks and exploit SNNs to process graph structure inputs [Xu *et al.*, 2021; Zhu *et al.*, 2022].

3 Neuromorphic Datasets

3.1 Recent Advances

In the DNNs field, the continuous expansion of datasets in image, text, and other areas poses a challenge to the performance of DNNs but also promotes the development of DNNs from another perspective. The same is true for SNNs. The datasets inspired by neuromorphic vision sensors' imaging mechanisms are called neuromorphic datasets and are considered the most suitable dataset type for SNNs' applications.

Neuromorphic vision sensors (NVSs) inspired by biological visual processing mechanisms mainly capture light intensity changes in the visual field. They record spike train information in positive and negative directions according to the direction of information change, making NVSs have the characteristics of low latency, asynchronous, and sparse. Representative NVSs include dynamic vision sensors and dynamic, active imaging sensors.

The following characteristics of neuromorphic datasets make them particularly suitable for benchmarking SNNs: 1) SNNs can naturally process asynchronous, event-driven information, making it a good fit with the data characteristics of neuromorphic datasets; 2) Temporal features embedded in neuromorphic datasets (such as precise firing times and temporal correlation between frames) provide an excellent platform to demonstrate the ability to spiking neurons to process information via spatially-temporal dynamics.

According to the dataset construction method, the current neuromorphic datasets are mainly divided into three categories. The first category is the datasets collected from the field scene, which are primarily captured directly by NVSs to generate unlabeled data, such as DvsGesture [Amir *et al.*, 2017] for gesture recognition. The second category is the transformation datasets, mainly generated from the labeled static image datasets through the actual shooting of NVSs, such as DVS-CIFAR10 [Li *et al.*, 2017]. Due to their ease of use and evaluation, such transformation datasets are the most commonly used datasets in SNNs. The third category is the generated datasets, which are mainly generated using labeled data through algorithms that simulate the characteristics of NVSs. They generate neuromorphic datasets directly from the existing image or video stream information through a specific difference algorithm [Bi and Andreopoulos, 2017].

3.2 New Frontiers

Although the research on neuromorphic datasets is still developing, these three categories of datasets have their limitations. Due to inconsistencies in the way researchers preprocessed the first category datasets, such as time resolution and image compression scale, the results reported so far are difficult to compare fairly. The second and third categories of datasets are mainly generated by the secondary transformation of the original static data, and their data is difficult to express rich temporal information. Therefore, they cannot fully use the spatially-temporal processing characteristics of SNNs. In summary, the current research on neuromorphic datasets is still in its infancy.

On the static image datasets in the DNNs field, e.g., MNIST [LeCun, 1998], CIFAR [Krizhevsky *et al.*, 2009], etc., the performance of SNNs is usually not as good as that of DNNs. However, studies have shown that it is unwise to blindly measure SNNs on such datasets with a single criterion, e.g., classification accuracy [Deng *et al.*, 2020]. In datasets that contain more dynamic temporal information and naturally have the form of spike signals, SNNs can achieve better results in terms of performance and computational overhead. As mentioned above, the small-scale datasets obtained by NVSs are the mainstream of current SNN datasets, but it does not rule out that there may be other more suitable data sources to be explored. In addition to the simple image recognition task, it is hoped to develop spatially-temporal event flow datasets ideal for diverse tasks to investigate further the potential advantages and possible application scenarios of SNNs. Moreover, constructing larger-scale and more functionally fit datasets (fully exploiting the spatially-temporal processing capabilities of spiking neurons and the event-driven properties of data) is also an important future direction to provide a broad and fair benchmark for SNNs.

4 Optimization Algorithms

4.1 Recent Advances

The learning of ANNs is to optimize network parameters based on task-specific datasets. Optimization algorithms play a crucial role in it. In DNNs, gradient-based error BP optimization algorithms [Rumelhart *et al.*, 1986] are the core of the current DNNs optimization theory and are widely used in practical scenarios. In contrast, there is no recognized core optimization algorithm in the SNNs field. There are different emphases between biological plausibility and task performance. In addition, different neuron models, encoding methods, and topological structures used in the network all lead to the diversification of optimization algorithms. The research on optimization algorithms of SNNs can be divided into two main types. One type is designed to understand better the biological system, where detailed biologically-realistic neural models are used without further consideration of computational performance. The other type is constructed to pursue superior computational performance, where only limited features of SNNs are retained, and some efficient but not biologically-plausible tuning algorithms are still used.

The first category of algorithms satisfies known BS discoveries as much as possible. This paper innovatively fur-

ther divides them into plasticity optimization based on micro-scale, meso-scale and macro-scale. Micro- and meso-scale plasticity are typically self-organizing, unsupervised local algorithms, and macro-scale plasticity is typically supervised global algorithms. Micro-scale plasticity mainly describes the properties of learning that take place at a single neuron or synaptic site, including STDP [Bi and Poo, 1998], short term plasticity (STP) [Zhang *et al.*, 2018b], Reward-STDP [Mozafari *et al.*, 2019], Dale rule [Yang *et al.*, 2016], etc. Such algorithms achieve decent performance on simple image classification tasks. Diehl [2015] uses two-layer SNNs with LIF neurons, and the adjacent layer neurons use STDP for learning, achieving a test accuracy of 95% on MNIST. It is subsequently optimized to 96.7% accuracy by incorporating plasticity mechanisms such as symmetric-STDP and dopamine modulation [Hao *et al.*, 2020]. Kheradpisheh [2018] uses multi-layer convolution, STDP, and information delay for efficient image feature classification, achieving 98.40% accuracy on MNIST. Moreover, a combined optimization algorithm of STDP and Reward-STDP is proposed to optimize multi-layer spiking convolutional networks [Mozafari *et al.*, 2019]. Meso-scale plasticity mainly describes the relationship between multiple synapses and multiple neurons, e.g., lateral inhibition [Zenke *et al.*, 2015], Self-backpropagation (SBP) [Zhang *et al.*, 2021b], homeostatic control among multiple neurons, etc. Zhang [2018a] proposes an optimization algorithm based on neural homeostasis to stabilize a single node's input and output information. Macro-scale plasticity mainly describes the top-down global credit distribution. Unfortunately, there is no global optimization algorithm similar to BP in the credit assignment of biological networks. The directionality of synaptic information transmission makes forward transmission and possible feedback pathways physiologically separate. The brain has no known way to access forward weights during backpropagation, which is called the weight transport problem. To make BP more biological-like and energy-efficient, some transformative algorithms for BP have emerged. For example, target propagation [Bengio, 2014], feedback alignment [Lillicrap *et al.*, 2016], direct random target propagation [Frenkel *et al.*, 2021], etc., solve the weight transport problem by implementing direct gradient transfer in the backward process with random matrices. They bring new ideas to the plasticity optimization of SNNs at the macro-scale, e.g., biologically-plausible reward propagation (BRP) [Zhang *et al.*, 2021c].

The second category of algorithms typically employs different BP-based variants for the optimization of SNNs, mainly including pseudo-BP [Zenke and Ganguli, 2018], DNNs-converted SNNs [Cao *et al.*, 2015], etc. Since the spike signal is not differentiable, the direct application of gradient-based BP is difficult. The key feature of the pseudo-BP is replacing the non-differential parts of spiking neurons during BP with a predefined gradient number [Shrestha and Orchard, 2018]. On some smaller-scale datasets, its performance and convergence speed are comparable to DNNs after standard BP training. The basic idea of DNNs-converted SNNs is that the average firing rate under rate encoding in SNNs can approximate the continuous activation value under the ReLU activation function in DNNs. After the orig-

inal DNNs are trained with BP, it is converted into SNNs by specific means [Cao *et al.*, 2015; Deng and Gu, 2020; Li and Zeng, 2022; Li *et al.*, 2022]. In terms of performance, the DNNs-converted SNNs maintain the smallest gap with DNNs and can be implemented on large-scale network structures and datasets. For example, Rueckauer [Rueckauer *et al.*, 2017] implements some spiking versions of the VGG-16 and GoogLeNet models. Sengupta [Sengupta *et al.*, 2019] reports that the VGG-16 achieved 69.96% accuracy on the ImageNet dataset with a conversion precision loss of 0.56%. Subsequently, Hu [Hu *et al.*, 2018] uses the deep structure of ResNet-50 to obtain 72.75% accuracy.

4.2 New Frontiers

The organic combination of biological plausibility and performance will remain the relentless goal of SNNs optimization algorithms. Compared with DNNs, only a few algorithms can directly train truly large-scale deep SNNs. Problems such as gradient vanishing, high resource overhead, and even non-convergence in deep networks training need further exploration. Recently, residual learning and batch normalization from the DNNs field has been introduced into pseudo-BP to train deep SNNs directly [Fang *et al.*, 2021; Hu *et al.*, 2021; Zheng *et al.*, 2021], which achieves excellent results and may serve as a way for the optimization development of deep SNNs in the future. Moreover, existing DNNs-converted SNNs algorithms also suffer from long simulation periods. From the perspective of model compression, the conversion process is an extreme quantization of activation values. The binary neural networks (BNNs) [Rastegari *et al.*, 2016; Chen *et al.*, 2018] in DNNs have a similar concept. However, the connection and difference between the BNNs and SNNs, and the possible impact of the additional temporal dimension in SNNs are not clearly elaborated. The threshold firing properties of SNNs may make them more receptive to compression algorithms. Therefore, the combination with compression algorithms such as weight quantization and pruning also needs to be explored so that the computational efficiency advantage of SNNs can be further developed [Chen *et al.*, 2021].

5 Software Frameworks

5.1 Recent Advances

The software framework of SNNs is a programming tool used to help the SNNs to achieve rapid simulation, network modeling, and algorithm training. Software frameworks are related to the effective reduction of the field entry threshold and the efficient development of large-scale SNNs projects, providing substantial support for the scientific research of SNNs. Due to the differences in research goals and implementation methods, there are many software frameworks.

Some software frameworks have been used to achieve smaller-scale neuronal functional simulations, mainly to understand biological systems. Neuron [Migliore *et al.*, 2006] and Nest [Gewaltig and Diesmann, 2007] are two commonly used software frameworks that support multiple programming languages, e.g., Python, C++, etc., and visual interfaces. They also support the characterization of more detailed neuronal activity dynamics, such as H-H, LIF, and Izhikevich, or

multi-compartmental models with complex structures. Other frameworks can implement task-specific, larger-scale SNNs optimization calculations, support multiple neuron models, and support various types of synaptic plasticity, such as STDP and STP. For example, Bindsnet [Hazan *et al.*, 2018], Brain2 [Stimberg *et al.*, 2019], Spyketorch [Mozafari *et al.*, 2019], SpikingJelly [Fang *et al.*, 2020], CogSNN [Zhang and Liu, 2022], etc., based on the Python language, better support multi-neuron networking, and can be used for some relative complex pattern recognition tasks. In particular, CogSNN [Zhang and Liu, 2022] has excellent support capabilities for neuromorphic datasets introduced earlier, such as N-MNIST, DVS-CIFAR10, and DvsGesture, and supports cognitive computations such as the Muller-Lyer illusion and McGurk effect.

5.2 New Frontiers

The software frameworks of SNNs are still in a relatively primary development stage. In DNNs, many kinds of software frameworks support their training, and the common one is PyTorch [Paszke *et al.*, 2019]. The user-friendly programming interface and the unified data stream processing method make it easy for a beginner to build and train DNNs, which significantly promotes the development of the DNNs field. However, in SNNs, currently, only a few frameworks can support the construction and training of large-scale SNNs. Building large SNNs still requires programmers to have excellent programming skills. Therefore, the development of user-friendly programming frameworks to effectively deploy large-scale SNNs is crucial to the development of this field.

6 Hardware Frameworks

6.1 Recent Advances

The development of the SNNs software frameworks enables the corresponding applications to be extended to more and more practical scenarios quickly. In particular, application scenarios with high demands for limited size, low energy consumption, parallel computing, etc., such as robot chips, high-performance analog computing, pattern recognition accelerators, and event high-speed cameras, have gradually begun to show great application potential.

Since SNNs have the advantage of ultra-low energy consumption in hardware circuits, in the last ten years, neuromorphic chips that support SNNs hardware implementation, represented by TrueNorth [Akopyan *et al.*, 2015], Loihi [Davies *et al.*, 2018], and Tianjic [Pei *et al.*, 2019] chips, have sprung up. Unlike the traditional Von Neumann processor architecture, many computing cores work simultaneously in a neuromorphic chip, exchanging intermediate results through a routing network. The whole system usually does not have a unified external memory. Instead, each computing core has its own independent storage space, presenting a decentralized operation mode, so it has incredibly high parallelism and memory access efficiency.

The existing neuromorphic chips can be divided into offline and online chips according to whether they support learning functions. Offline chips mean that the parameters (e.g., weights) of SNNs have been trained in advance. The model only needs to be deployed to the neuromorphic chips, and the parameters will not be updated in the subsequent running process. That said, the offline chips only support the inference process of SNNs, but not their training process. Such chips include TrueNorth [Akopyan *et al.*, 2015], Tianjic [Pei *et al.*, 2019], and Neurogrid [Benjamin *et al.*, 2014]. Unlike the offline chips, the online chips support parameter updates during the running process of the SNNs model. Such chips include Loihi [Davies *et al.*, 2018], SpiN-Naker chip [Furber *et al.*, 2014], and some developing chips equipped with CogSNN toolbox [Zhang and Liu, 2022].

IBM TrueNorth chip [Akopyan *et al.*, 2015] contains about 5.4 billion silicon transistors, 4096 cores, 1 million neurons, and 256 million synapses, which can realize applications such as SNNs-based brain-inspired affective computing. The number of neurons in the Intel Loihi chip has reached 8 million, and synapses have reached 8 billion [Davies *et al.*, 2018]. It initially plays a role in highly sensitive odor perception and recognition. Stanford University Neurogrid chip simulates millions of neurons connected by billions of synapses in real-time, supporting high-performance computers and brain-like robot chips [Benjamin *et al.*, 2014]. Its current progress includes high-throughput brain information processing, brain-computer interface neural information recording, etc. The DNNs/SNNs hybrid brain-inspired Tianjic chip developed by Tsinghua University can support traditional DNNs, and a new generation of SNNs [Pei *et al.*, 2019]. They also verified the functions of speech recognition, control tracking, and automatic obstacle avoidance of Tianjic on self-driving bicycles.

6.2 New Frontiers

In the context of the Von Neumann bottleneck, as an alternative computing paradigm to traditional digital circuits, neuromorphic chips have become a research hotspot for more than ten years and have achieved fruitful results. By drawing inspiration from the brain's structure and function, neuromorphic chips provide an efficient solution for event-driven computation in SNNs, achieving essential properties such as high parallelism and ultra-low power consumption. Combining various advantageous technologies of existing hardware chips is an important direction that requires in-depth study. This cross-integration may be reflected in the following aspects: 1) Heterogeneous fusion of two paradigms, DNNs and SNNs, improves overall performance; 2) Mixed-precision computation of low-precision memristors and high-precision digital circuits; 3) A general-purpose computing chip combines high-efficiency but low-performance unsupervised local learning with high-performance but low-efficiency supervised global learning.

7 Conclusion

In this paper, we provide a literature survey for SNNs. We review the recent advances and discuss the new frontiers in SNNs from five major research topics: essential elements (i.e., neuron models, encoding methods, and topology structures), neuromorphic datasets, optimization algorithms, software, and hardware frameworks. We hope that this survey can shed light on future research in the SNNs field.

Acknowledgments

References

[Adrian and Zotterman, 1926] Edgar D Adrian and Yngve Zotterman. The impulses produced by sensory nerve-endings: Part II. The response of a Single End-Organ. *The Journal of Physiology*, 61(2):151, 1926.

[Akopyan *et al.*, 2015] Filipp Akopyan, Jun Sawada, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *TCAD*, 34(10):1537–1557, 2015.

[Amir *et al.*, 2017] Arnon Amir, Brian Taba, et al. A low power, fully event-based gesture recognition system. In *CVPR*, pages 7243–7252, 2017.

[Bengio, 2014] Yoshua Bengio. How auto-encoders could provide credit assignment in deep networks via target propagation. *arXiv preprint arXiv:1407.7906*, 2014.

[Benjamin *et al.*, 2014] Ben Varkey Benjamin, Peiran Gao, et al. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014.

[Bi and Andreopoulos, 2017] Yin Bi and Yiannis Andreopoulos. PIX2NVS: Parameterized conversion of pixel-domain video frames to neuromorphic vision streams. In *ICIP*, pages 1990–1994. IEEE, 2017.

[Bi and Poo, 1998] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24):10464–10472, 1998.

[Cao *et al.*, 2015] Yongqiang Cao, Yang Chen, et al. Spiking deep convolutional neural networks for energy-efficient object recognition. *IJCV*, 113(1):54–66, 2015.

[Chen *et al.*, 2018] Xiuyi Chen, Guangcan Liu, et al. Distilled Binary Neural Network for Monaural Speech Separation. In *IJCNN*, pages 1–8, 2018.

[Chen *et al.*, 2021] Yanqi Chen, Zhaofei Yu, et al. Pruning of Deep Spiking Neural Networks through Gradient Rewiring. In *IJCAI*, 2021.

[Cheng *et al.*, 2020] Xiang Cheng, Yunzhe Hao, et al. LISNN: Improving Spiking Neural Networks with Lateral Interactions for Robust Object Recognition. In *IJCAI*, pages 1519–1525, 2020.

[Davies *et al.*, 2018] Mike Davies, Narayan Srinivasa, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[Dayan *et al.*, 2003] Peter Dayan, Laurence F Abbott, et al. Theoretical neuroscience: computational and mathematical modeling of neural systems. *Journal of Cognitive Neuroscience*, 15(1):154–155, 2003.

[Deng and Gu, 2020] Shikuang Deng and Shi Gu. Optimal Conversion of Conventional Artificial Neural Networks to Spiking Neural Networks. In *ICLR*, 2020.

[Deng *et al.*, 2020] Lei Deng, Yujie Wu, et al. Rethinking the performance comparison between SNNS and ANNS. *Neural Networks*, 121:294–307, 2020.

[Diehl and Cook, 2015] Peter U Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational neuroscience*, 9:99, 2015.

[Fang *et al.*, 2020] Wei Fang, Yanqi Chen, et al. SpikingJelly. https://github.com/fangwei123456/spikingjelly, 2020. Accessed: 2022-05-02.

[Fang *et al.*, 2021] Wei Fang, Zhaofei Yu, et al. Deep residual learning in spiking neural networks. *NeurIPS*, 34, 2021.

[Frenkel *et al.*, 2021] Charlotte Frenkel, Martin Lefebvre, et al. Learning without feedback: Fixed random learning signals allow for feedforward training of deep neural networks. *Frontiers in Neuroscience*, page 20, 2021.

[Furber *et al.*, 2014] Steve B Furber, Francesco Galluppi, et al. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014.

[Georgopoulos *et al.*, 1986] Apostolos P Georgopoulos, Andrew B Schwartz, et al. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

[Gewaltig and Diesmann, 2007] Marc-Oliver Gewaltig and Markus Diesmann. Nest (neural simulation tool). *Scholarpedia*, 2(4):1430, 2007.

[Hao *et al.*, 2020] Yunzhe Hao, Xuhui Huang, et al. A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule. *Neural Networks*, 121:387–395, 2020.

[Hazan *et al.*, 2018] Hananel Hazan, Daniel J Saunders, et al. Bindsnet: A machine learning-oriented spiking neural networks library in python. *Frontiers in Neuroinformatics*, page 89, 2018.

[Hinton *et al.*, 2006] Geoffrey E Hinton, Simon Osindero, et al. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[Hodgkin and Huxley, 1952] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500, 1952.

[Hu *et al.*, 2018] Yangfan Hu, Huajin Tang, et al. Spiking Deep Residual Networks. *IEEE TNNLS*, 2018.

[Hu *et al.*, 2021] Yifan Hu, Yujie Wu, et al. Advancing Deep Residual Learning by Solving the Crux of Degradation in Spiking Neural Networks. *arXiv preprint arXiv:2201.07209*, 2021.

[Izhikevich *et al.*, 2004] Eugene M Izhikevich, Joseph A Gally, et al. Spike-timing dynamics of neuronal groups. *Cerebral Cortex*, 14(8):933–944, 2004.

[Jia *et al.*, 2022] Shuncheng Jia, Ruichen Zuo, et al. Motif-topology and Reward-learning improved Spiking Neural Network for Efficient Multi-sensory Integration. *ArXiv preprint arXiv:2202.06821*, 2022.

[Kheradpisheh *et al.*, 2018] Saeed Reza Kheradpisheh, Mohammad Ganjtabesh, et al. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99:56–67, 2018.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[LeCun, 1998] Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[Li and Zeng, 2022] Yang Li and Yi Zeng. Efficient and Accurate Conversion of Spiking Neural Network with Burst Spikes. *arXiv preprint arXiv:2204.13271*, 2022.

[Li *et al.*, 2017] Hongmin Li, Hanchao Liu, et al. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in Neuroscience*, 11:309, 2017.

[Li *et al.*, 2022] Yang Li, Xiang He, et al. Spike Calibration: Fast and Accurate Conversion of Spiking Neural Network for Object Detection and Segmentation. *arXiv preprint arXiv:2207.02702*, 2022.

[Lillicrap *et al.*, 2016] Timothy P Lillicrap, Daniel Cownden, et al. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7(1):1–10, 2016.

[Maass, 1997] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[Migliore *et al.*, 2006] Michele Migliore, C Cannia, et al. Parallel network simulations with NEURON. *Journal of Computational Neuroscience*, 21(2):119–129, 2006.

[Mozafari *et al.*, 2019] Milad Mozafari, Mohammad Ganjtabesh, et al. Spyketorch: Efficient simulation of convolutional spiking neural networks with at most one spike per neuron. *Frontiers in Neuroscience*, page 625, 2019.

[Orchard *et al.*, 2015] Garrick Orchard, Ajinkya Jayawant, et al. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9:437, 2015.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019.

[Pei *et al.*, 2019] Jing Pei, Lei Deng, et al. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

[Rastegari *et al.*, 2016] Mohammad Rastegari, Vicente Ordonez, et al. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, pages 525–542. Springer, 2016.

[Rosenblatt, 1958] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.

[Rueckauer *et al.*, 2017] Bodo Rueckauer, Iulia-Alexandra Lungu, et al. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11:682, 2017.

[Rumelhart *et al.*, 1986] David E Rumelhart, Geoffrey E Hinton, et al. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[Sengupta *et al.*, 2019] Abhronil Sengupta, Yuting Ye, et al. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in Neuroscience*, 13:95, 2019.

[Shrestha and Orchard, 2018] Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *NeurIPS*, 31, 2018.

[Sporns *et al.*, 2004] Olaf Sporns, Rolf Kötter, et al. Motifs in brain networks. *PLoS Biology*, 2(11):e369, 2004.

[Stimberg *et al.*, 2019] Marcel Stimberg, Romain Brette, et al. Brian 2, an intuitive and efficient neural simulator. *Elife*, 8:e47314, 2019.

[Thorpe and Gautrais, 1998] Simon Thorpe and Jacques Gautrais. Rank order coding. In *Computational Neuroscience*, pages 113–118. Springer, 1998.

[VanRullen *et al.*, 2005] Rufin VanRullen, Rudy Guyonneau, et al. Spike times make sense. *Trends in Neurosciences*, 28(1):1–4, 2005.

[Xu *et al.*, 2021] Mingkun Xu, Yujie Wu, et al. Exploiting Spiking Dynamics with Spatial-temporal Feature Normalization in Graph Learning. *arXiv preprint arXiv:2107.06865*, 2021.

[Yang *et al.*, 2016] Guangyu Robert Yang, John D Murray, et al. A dendritic disinhibitory circuit mechanism for pathway-specific gating. *Nature Communications*, 7(1):1–14, 2016.

[Zenke and Ganguli, 2018] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, 30(6):1514–1541, 2018.

[Zenke *et al.*, 2015] Friedemann Zenke, Everton J Agnes, et al. Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nature Communications*, 6(1):1–13, 2015.

[Zhang and Liu, 2022] Tielin Zhang and Hongxing Liu. CogSNN. https://github.com/thomasaimondy/CogSNN, 2022. Accessed: 2022-05-02.

[Zhang *et al.*, 2018a] Tielin Zhang, Yi Zeng, et al. A plasticity-centric approach to train the non-differential spiking neural networks. In *AAAI*, 2018.

[Zhang *et al.*, 2018b] Tielin Zhang, Yi Zeng, et al. Brain-inspired Balanced Tuning for Spiking Neural Networks. In *IJCAI*, pages 1653–1659. Stockholm, 2018.

[Zhang *et al.*, 2021a] Duzhen Zhang, Tielin Zhang, et al. Population-coding and Dynamic-neurons improved Spiking Actor Network for Reinforcement Learning. *ArXiv preprint arXiv:2106.07854*, 2021.

[Zhang *et al.*, 2021b] Tielin Zhang, Xiang Cheng, et al. Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks. *Science Advances*, 7(43):eabh0146, 2021.

[Zhang *et al.*, 2021c] Tielin Zhang, Shuncheng Jia, et al. Tuning convolutional spiking neural network with biologically plausible reward propagation. *IEEE TNNLS*, 2021.

[Zhang *et al.*, 2021d] Tielin Zhang, Yi Zeng, et al. Neuron type classification in rat brain based on integrative convolutional and tree-based recurrent neural networks. *Scientific Reports*, 11(1):1–14, 2021.

[Zheng *et al.*, 2021] Hanle Zheng, Yujie Wu, et al. Going Deeper With Directly-Trained Larger Spiking Neural Networks. In *AAAI*, volume 35, pages 11062–11070, 2021.

[Zhu *et al.*, 2022] Zulun Zhu, Jiaying Peng, Jintang Li, Liang Chen, Qi Yu, and Siqiang Luo. Spiking Graph Convolutional Networks. *arXiv preprint arXiv:2205.02767*, 2022.