Literature Review: Spike-based Machine Intelligence and Neuromorphic Computing

Chengting Yu
ZJU-UIUC Institute, Zhejiang University
chengting.17@intl.zju.edu.cn

***Abstract*—Recent years have seen the rapid development of the modern deep-learning networks (DLNs) in various aspects. Today's computer have demonstrated extraordinary abilities in several cognition tasks. However, the fact is that the present computing models are still quite different compared with the computing principles of the brain in biology. Therefore, with the deeper exploration in Neuromorphic Computing, the Spike-based Machine Intelligence has been a promising field to research further. The purpose of this study is to review the developments of spike-based machine intelligence in both algorithmic and hardware domains. The study summarizes several recent works, analyzing their contributions as well as limitations in each case, providing the brief overview of neuromorphic computing.**

***Keywords*—Neuromorphic Computing, Spiking Neural Network, Spiking-neuron Integrated Circuit**

I. Introduction

Inspired by the hierarchical structure of brain and neuro-synaptic framework, modern deep-learning networks (DLNs) are artefacts of hierarchy with composing several layers or transformations that represent different latent features in the input [2]. In fact, such neural networks are fuelled by hardware computing systems that fundamentally rely on basic silicon transistors, where various silicon-based computational aspects are arranged in a hierarchical fashion, like the hierarchical organization of the brain, to allow efficient data exchange. However, the silicon-based computers are still quite different compared with the computing principles of the brain, including the computing and storage mechanisms, the two- vs. three-dimensions in connectivity, the digital circuits vs. the spike-based event-driven computations in the brains [3], etc.

Guided by the neuromorphology,the spike-driven computations are being actively explored by algorithm designers to drive scalable, energy-efficient 'spiking neural networks' (SNNs). This review will go through the developments of spike-based artificial intelligence in both algorithmic and hardware domains, and review the particular articles in different areas to figure out their contributions as well as limitations.

II. Spiking neural networks

The categories of different generations of neural networks are firstly mentioned by Wolfgang Maass in 1996 [4], which further provided a rigorous mathematical analysis of the third generation of network with spiking neurons, showing the better computational power of the third network model as well as the theoretically fewer number of neurons needed to achieve some concrete functions, compared with the first two generations.

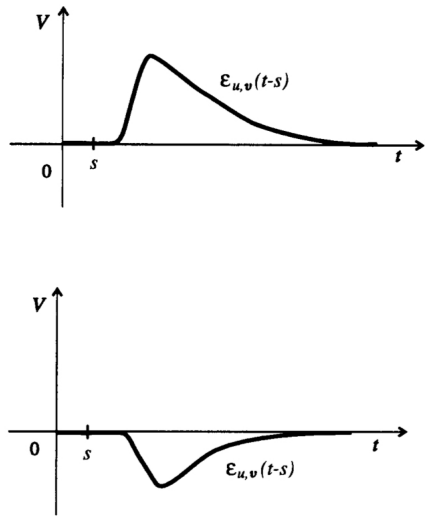

Fig. 1. Typical shape of response functions (EPSP and IPSP) of a biological neuron.

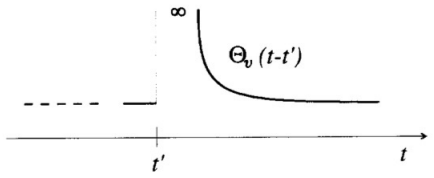

Fig. 2. Typical shape of the threshold function of a biological neuron.

The paper, *Networks of spiking neurons: the third generation of neural network models [4]*, provides several theorems following with rigorous mathematical proof about the spiking neural network. However, the theorems are only adapted in the special cases with concrete functions, which might need to be integrated and abstract to the more general formal to help guide the spike-based machine intelligence in both algorithmic and hardware domains. Still, we can glimpse the potential of spiking neuron model, especially in computationally efficient intelligence applications such as recognition and inference.

III. Learning Algorithm for SNN

The advantage of spiking neural network is its high bio-explanatory. On one hand, SNN can be used as a basic

tool for computational neurology to simulate biological brain phenomena. On the other hand, the spike-based structure is easier to be implemented on hardware, such as on-chip systems like FPGA, due to its special characteristics of information transmission. However, since the impulse function is not differentiable, the gradient-descent method cannot be directly applied to train the networks, remaining the learning algorithm for SNN to be a major research problem in recent years.

The state-of-the-art technologies on learning of spiking neural network can be divided into two aspects – Conversion-based approaches and Spike-based approaches.

A. Conversion-based approaches

Conversion-based approaches, or "rate-based learning" [11], focus on the transformation from the pre-existing DLN into corresponding SNN, which yields the same input-output mapping for a given task. Such conversion-based approaches avoid the training process of SNNs in the temporal domain and have flexibility on the expansion of the existing DLN models.

In the paper, *Going deeper in spiking neural networks: VGG and residual architectures [5]*, the group proposed a new concrete ANN-SNN conversion technique with better performance, called Spike-Norm, and described the insights and design constraints in ANN-SNN conversion of Residual Network, which is a potential pathway to enable deeper SNNs.

Table 1. Results for CIFAR-10 Dataset

Network Architecture	ANN Error	SNN Error	Error Increment
3-layered networks (Esser et al., 2016)	-	10.68%	-
8-layered networks (Hunsberger & Eliasmith, 2016)	16.28%	16.46%	0.18%
VGG-16 (ANN model based conversion)	8.3%	8.54%	0.24%
VGG-16 (SPIKE-NORM)	**8.3%**	**8.45%**	**0.15%**

Table 2. Results for ImageNet Dataset

Network Architecture	ANN Error	SNN Error	Error Increment
VGG-16 (ANN model based conversion)	29.48% (10.61%)	30.61% (11.21%)	1.13% (0.6%)
VGG-16 (SPIKE-NORM)	**29.48% (10.61%)**	**30.04% (10.99%)**	**0.56% (0.38%)**

Table 3. Results for Residual Networks

Dataset	Network Architecture	ANN Error	SNN Error
CIFAR-10	ResNet-20	10.9%	12.54%
ImageNet	ResNet-34	29.31% (10.31%)	34.53% (13.67%)

Fig. 3. The results of conversion-based SNNs

This paper was the first to demonstrate the competitive performance of a conversion-based spiking neural network on ImageNet data for deep neural architectures, with the error competition shown in Fig.3. However, it is still an open area of exploration on the improvement of SNNs performance with Conversion-based approaches. The conversion constraints like bias, max-pooling, etc. should be explored further, while the accuracy loss on conversion should be reduced further.

B. Spike-based approaches

Spike-based approaches, focus on the direct training of SNNs with using temporal and event-based information, which offers sparsity and efficiency in overall spiking dynamics. Unsupervised train and supervised train are two main directions in spike-based approaches. Those methods often rely on different variants of models of STDP (shown in Fig.4), giving a closer behavior to neurons in biology.

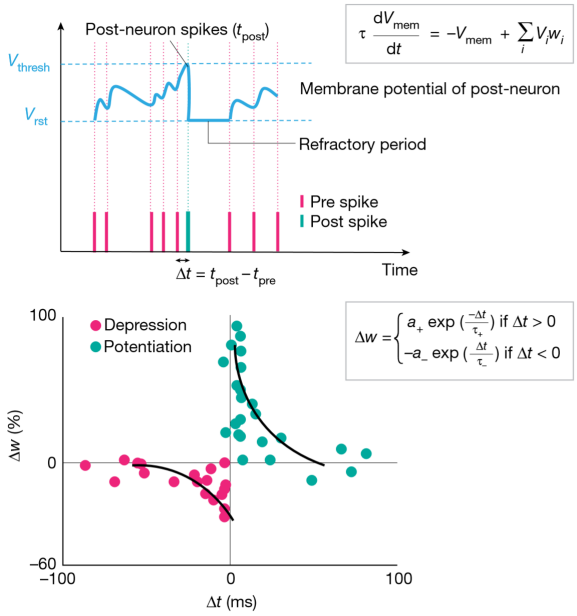

Fig. 4. The leaky integrate-and-fire (LIF) model and the spike-timing-dependent plasticity (STDP) formulation.

Early works in supervised learning were based on simple spike-based learning in a single-layer SNN using STDP to perform classification, such as ReSuMe [6] and the tempotron [7]. Then, many researches explored the backpropogation method to enable supervised learning in multi-layer SNNs. Most of them estimate a differentiable approximate function for the spiking neuronal functionality in order to perform gradient-descent method. SpikeProp [8] and related variants [9,10] have derived a backpropagation rule for SNNs by fixing a target spike train at the output layer. Recent works achieved the deep SNNs for small-scale image recognition tasks with more computational efficiency, however, those supervised-learning algorithms did still not outperform conversion-based approaches in terms of accuracy for large-scale tasks.

Unsupervised training method is also considerable in spike-based approaches. Diehl and Cook, in their paper, *Unsupervised learning of digit recognition using spike-timing-dependent plasticity [11]*, provided a completely STDP-based

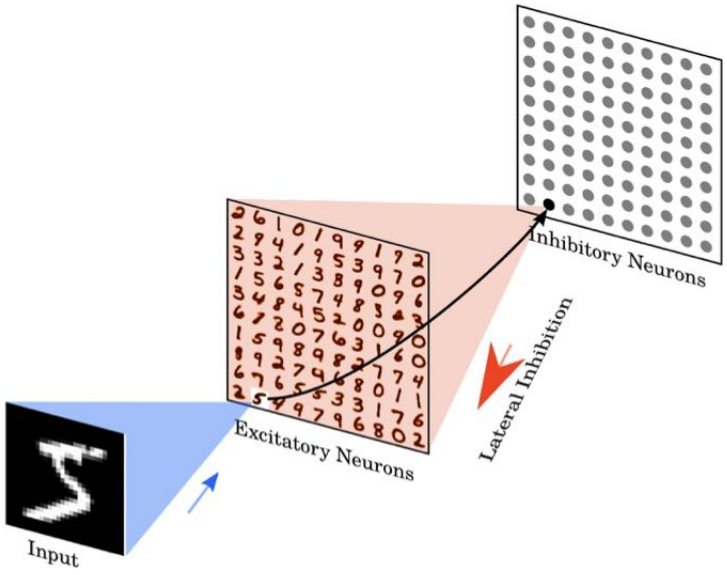

FIGURE 1 | Network architecture. The intensity values of the 28 × 28 pixel MNIST image are converted to Poisson-spike with firing rates proportional to the intensity of the corresponding pixel. Those Poisson-spike trains are fed as input to excitatory neurons in an all-to-all fashion. The blue shaded area shows the input connections to one specific excitatory example neuron. Excitatory neurons are connected to inhibitory neurons via one-to-one connections, as shown for the example neuron. The red shaded area denotes all connections from one inhibitory neuron to the excitatory neurons. Each inhibitory neuron is connected to all excitatory neurons, except for the one it receives a connection from. Class labels are not presented to the network, so the learning is unsupervised. Excitatory neurons are assigned to classes after training, based on their highest average response to a digit class over the training set. No additional parameters are used to predict the class, specifically no linear classifier or similar methods are on top of the SNN.

Fig. 5. The unsupervised-learning model given by Diehl at el.

unsupervised learning on SNNs (Fig.5), which gives a comparable accuracy to deep learning on the MNIST database. However, there are still many gaps in the research on the unsupervised train of SNNs, where the complex architecture, the computational efficiency and the overall accuracy are still waited to be explored in the future works.

IV. Hardware design for Neuromorphic Computing

With the developments of neuromorphic computing, we gradually perceived the difference between the architecture of the brain and the von Neumann architecture of present computers. In fact, while the brain is parallel and distributed architecture in the nature, the today's computer, however, is sequential and centralized architecture called "von Neumann", which suffers the bottleneck caused by separation between storage and computation units. Under the circumstances, some of recent works explored the brand-new architecture of neuromorphic computing models that allow to "beyond von Neumann", or even "beyond silicon". To achieve either "near-memory" computing or "in-memory" computing is the most promising direction in mitigating the limitation of the memory wall bottleneck to go beyond von Neumann [12,13].

A. Near-memory computing

Near-memory computing. It allows co-location of memory and computing by embedding a dedicated processing engine in close proximity to the memory unit. This concept is consistent among the distributed computing architecture of various spike-based "Big Brain chips", such as Neuogrid and TrueNorth (Fig.6).

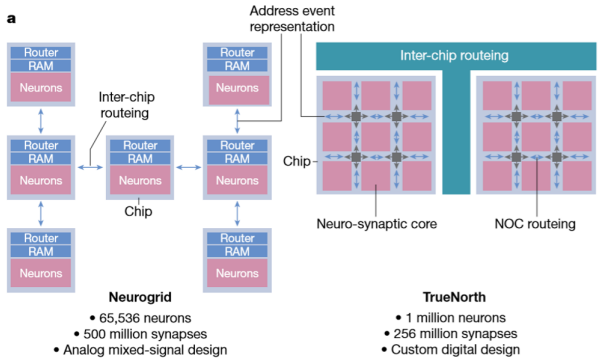

Fig. 6. The representative spike-based 'Big Brain' chips.

The key architectural abstraction of TrueNorth is shown in Fig.8. Inspired by neuroscience, TrueNorth is a distributed network of neurosynaptic cores that allows the digital neuromorphic implementation of the large-scale SNNs that are efficient, scalable, and flxible within today's technology [14]. It is the first digital custom-designed, large-scale neuromorphic processor, an outcome of the DARPA SyNAPSE pro-

gramme, providing a sample for computing models "beyond von Neumann". The concrete description of architecture and experiment results could be found in the paper, A million spiking-neuron integrated circuit with a scalable communication network and interface. Although TrueNorth significantly improves the traditional hardware architecture, it still has a gap with our brain structure. One certain flaw is that, it does not consider the synaptic efficacy as well as synaptic plasticity of neurons, missing the STDP rules in neuromorphic computing.

B. In-memory computing.

In-memory computing. It embeds certain aspects of computational operations within the memory array by enabling computation in the memory bit-cells or the peripheral circuits, which is a promising aspect to break through the bottleneck of the memory wall, making the architecture model closer to the real brain in biology. The almost recent works of in-memory computing focus on the major memory technologies, including static and dynamic silicon memories, and non-volatile memristive technologies (RRAMs, PCMs and STT-MRAMs) as well.

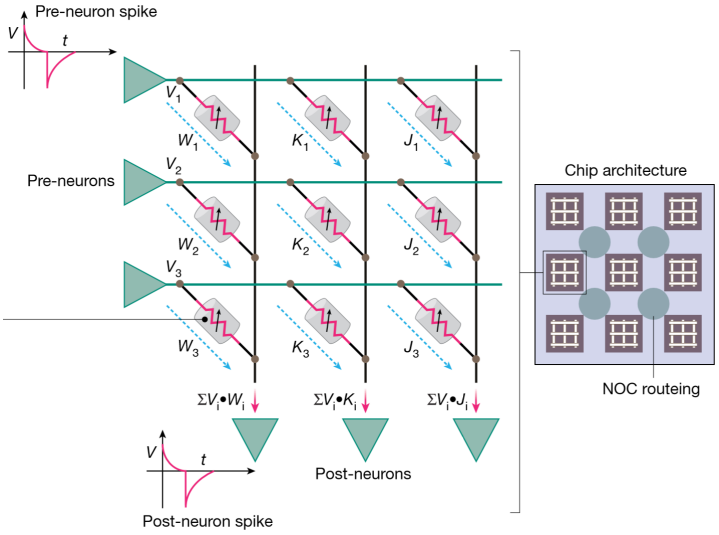

Fig. 7. The implementation of synaptic efficacy and plasticity using memristive technologies

Using non-volatile memristive technologies with arranging in a crossbar fashion (Fig.7), the computing model is enable to accomplished in situ synaptic efficacy and synaptic plasticity, which are two of the most important characteristics of biological synapses. The memristive technologies achieve the synaptic plasticity by appropriately applying voltage pulses, corresponding to the learning rule of STDP. Under the circumstances, there is a great potential for combining the spike-based computing architecture, such as TrueNorth, with the memristive technologies to create a better hardware architecture with allowing the synaptic efficacy and synaptic plasticity.

V. Conclusion

In this review, we discussed both algorithmic and hardware aspects of Spike-based Machine Intelligence. We began by briefly introducing the mathematical basis of Spiking Neural Network, showing its great potential in both energy and space. We then described two main learning approaches of SNNs (Conversion-based and Spike-based approaches), showed the corresponding works in each aspect, and further discussed their results, contributions and limitations. Finally, we reviewed the developments of recent hardware designs, analyzing their relations and insufficiency in Neuromorphic Computing. Nowadays, Spike-based Machine Intelligence still has huge potential to be tapped. With more and more research input, I believe the Spike-based Machine Intelligence will eventually achieve a breakthrough in artificial intelligence for human and become an important step to uncover the mysteries of brain and consciousness.

References

[1] Roy, K., Jaiswal, A. & Panda, P. Towards spike-based machine intelligence with neuromorphic computing. Nature 575, 607–617 (2019).

[2] Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems Vol. 28 (eds Pereira, F. et al.) 1097–1105 (Neural Information Processing Systems Foundation, 2012).

[3] Deco, G., Rolls, E. T. & Romo, R. Stochastic dynamics as a principle of brain function. Prog. Neurobiol. 88, 1–16 (2009).

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

[5] Sengupta, A., Ye, Y., Wang, R., Liu, C. & Roy, K. Going deeper in spiking neural networks: VGG and residual architectures. Front. Neurosci. 13, 95 (2019).

[6] Ponulak, F. & Kasiński, A. Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting. Neural Comput. 22, 467–510 (2010).

[7] Gütig, R. & Sompolinsky, H. The tempotron: a neuron that learns spike-timing-based decisions. Nat. Neurosci. 9, 420–428 (2006).

[8] Bohte, S. M., Kok, J. N. & La Poutré, H. Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48, 17–37 (2002).

[9] Ghosh-Dastidar, S. & Adeli, H. A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. Neural Netw. 22, 1419–1431 (2009).

[10] Anwani, N. & Rajendran, B. NormAD: normalized approximate descent-based supervised learning rule for spiking neurons. In Int. Joint Conf. on Neural Networks 2361–2368 (IEEE, 2015).

[11] Diehl, P. U. & Cook, M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Front. Comput. Neurosci. 9, 99 (2015).

[12] Gokhale, M., Holmes, B. & Iobst, K. Processing in memory: the Terasys massively parallel PIM array. Computer 28, 23–31 (1995).

[13] Elliott, D., Stumm, M., Snelgrove, W. M., Cojocaru, C. & McKenzie, R. Computational RAM: implementing processors in memory. IEEE Des. Test Comput. 16, 32–41 (1999).

[14] Merolla, P. A. et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science 345, 668–673 (2014).

[15] Chua, L. Memristor—the missing circuit element. IEEE Trans. Circuit Theory 18, 507–519 (1971).

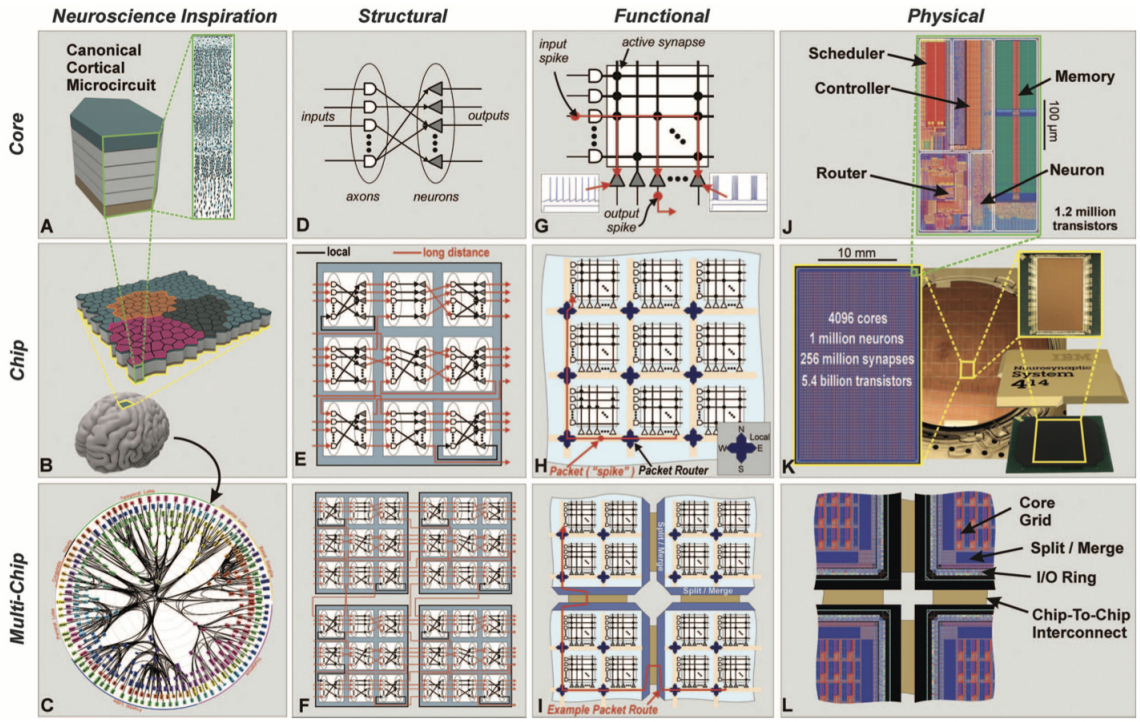

Fig. 2. TrueNorth architecture. Panels are organized into rows at three different scales (core, chip, and multichip) and into columns at four different views (neuroscience inspiration, structural, functional, and physical). (**A**) The neurosynaptic core is loosely inspired by the idea of a canonical cortical microcircuit. (**B**) A network of neurosynaptic cores is inspired by the cortex's two-dimensional sheet. (**C**) The multichip network is inspired by the long-range connections between cortical regions shown from the macaque brain (*30*). (**D**) Structure of a neurosynaptic core with axons as inputs, neurons as outputs, and synapses as directed connections from axons to neurons. Multicore networks at (**E**) chip scale and (**F**) multichip scale are both created by connecting a neuron on any core to an axon on any core with point-to-point connections. (**G**) Functional view of core as a crossbar where horizontal lines are axons, cross points are individually programmable synapses, vertical lines are neuron inputs, and triangles are neurons. Information flows from axons via active synapses to neurons. Neuron behaviors are individually programmable, with two examples shown. (**H**) Functional chip architecture is a two-dimensional array of cores where long-range connections are implemented by sending spike events (packets) over a mesh routing network to activate a target axon. Axonal delay is implemented at the target. (**I**) Routing network extends across chip boundaries through peripheral merge and split blocks. (**J**) Physical layout of core in 28-nm CMOS fits in a 240-µm-by-390-µm footprint. A memory (static random-access memory) stores all the data for each neuron, a time-multiplexed neuron circuit updates neuron membrane potentials, a scheduler buffers incoming spike events to implement axonal delays, a router relays spike events, and an event-driven controller orchestrates the core's operation. (**K**) Chip layout of 64-by-64 core array, wafer, and chip package. (**L**) Chip periphery to support multichip networks. I/O, input/output.

Fig. 8. The hierarchical architecture of TrueNorth