Chinese Journal of Electronics
Vol.27, No.4, July 2018

A Survey of Neuromorphic Computing Based on Spiking Neural Networks*

ZHANG Ming, GU Zonghua and PAN Gang

(*School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

Abstract — Neuromorphic computing aims to build digital or analog computer systems that emulate or simulate the biological brain, in order to achieve high performance and low power consumption for intelligent information processing applications. This article reviews on neuromorphic computing based on Spiking neural networks (SNNs), including its history of development, common neuron models, major research projects, neuromorphic sensors, and applications in brain-computer Interfaces.

Key words — Neuromorphic computing, Spiking neural networks, Brain-inspired computing, Machine learning.

I. Introduction

The von Neumann architecture, proposed by the mathematician John von Neumann in 1945, is the prevailing processor architecture today. Moore's law, which states that the number of transistors on a chip roughly doubles every 18 months, has been driving the exponential performance improvement of processor chips for the past 50 years. As the size of transistor continues to shrink, Dennard scaling, which states that power density of transistors roughly stays constant as they get smaller, stopped working around the year 2004. Processor clock speed stopped increasing due to memory wall and heat dissipation issues, and processor designers turned to multicore and manycore designs, in order to continue improving performance while maintaining an acceptable power density. Still, for large manycore chips, the high power consumption prevents most cores from running simultaneously, *i.e.*, the dark silicon problem. Researchers[1] believe that in near future, multicore scaling will come to an end due to the high power consumption, *i.e.*, it is no longer feasible to improve performance by increasing the number of processor cores on a single chip. On the other hand, as transistors get close to their physical limits, it is widely believed that Moore's law will no longer be in effect around 2020 due to both technical and economical reasons[2]. How will the semiconductor industry develop in the post Moore's-law era? Researchers from different fields have different answers to this question, including new materials, new devices, and new computer architectures.

Compared with computers, human brains perform poorly in numerical computation of exact values, even worse than a cheap calculator. However, human brains outperform computers in intelligent information processing tasks, such as object recognition, video/audio understanding, and natural language processing. In 1997, the human world chess champion Garry Kasparov was defeated by the IBM supercomputer DeepBlue, which employed hundreds of processors to perform brute-force search to find optimal moves. Thanks to the rise of big data, artificial intelligence, especially deep learning, has been making rapid progress. In 2016, Google's AlphaGo beat Lee Sedol, the World Champion in the Go game, based on deep learning and Monte Carlo tree search. In this article, we discuss neuromorphic computing (or neuromorphic engineering), including its history, neuron models in Spiking neural networks (SNNs), training algorithms, and neuromorphic sensors.

II. History

The main idea of neuromorphic computing based on neural network simulation, is to apply neural network principles to computer systems design, in order to achieve high performance and low power consumption for intelligent information processing applications. In the von Neumann architecture, computation and memory are separated into different physical units. This separation allows for software programmability, *i.e.*, a computer can be programmed with different softwares to carry out different

*Manuscript Received Mar. 14, 2017; Accepted July 4, 2017. This work is supported by National Key Basic Research and Development Program of China (No.2017YFB1002503) and the National Natural Science Foundation of China (No.61672454).
 DOI:10.1049/cje.2018.05.006

tasks. A drawback of this separation is that the communication delay between the computation unit and the memory unit may become the performance bottleneck, causing the memory wall problem. In contrast, computation and memory are unified in biological neural networks, hence it does not have the memory wall issue. The human brain is an extremely efficient computer with many advantages, including: self-learning by interaction with the environment (no need for explicit programming); highly fault-tolerant (tolerant to loss of neurons); massive parallelism (about 10^{11} neurons); massive connectivity (about 10^{15} synapses); low operating frequency (about 100Hz); slow signal transmission (several meters per second); and low power consumption (about 20W). Grace *et al.* proposed the number of Traversed edges per second (TEPS) on a large random graph as a metric for performance comparison between human brains and computers[3]. The key performance bottleneck of the biological neural network is communication among neurons instead of computation within neurons. The TEPS metric is based on the premise that transmitting a spike between neurons resembles traversing an edge of a large random graph. By the TEPS metric, a human brain is roughly 30 times faster than the fastest super computer (in 2015).

Hardware acceleration of machine learning algorithms, especially Artificial neural networks (ANNs), has a long history. Sze[4] gives a comprehensive tutorial and survey on this topic. Unlike traditional ANNs, biological neural networks communicate via discrete spikes instead of numerical values, forming SNNs. In an SNN (Spiking neural network), a neuron is activated only when it receives an input spike, hence inactive neurons without any input spikes can be put into low-power mode to save power. Therefore, SNNs can potentially achieve extreme low power consumption compared to ANNs, especially with implementation of Analog/mixed signal (AMS) circuits. In this survey, we mainly focus on research work on SNNs. Furthermore, SNNs' biological plausibility may help to achieve better integration of computing components and biological components in the quest for the long-term vision of cyborg intelligence[5−7].

We briefly review the history of neuromorphic computing:

1) In 1989, Carver Mead coined the term of Neuromorphic engineering, and proposed to simulate SNNs using sub-threshold analog circuits.

2) In 2004, Kwabena Boahen, a former student of Carver Mead, developed Neurogrid, a neuromorphic chip based on AMS circuits.

3) In 2005, the SpiNNaker (a contraction of spiking neural network architecture) project, based on multicore ARM processors, was started at the University of Manchester; The FACETS (Fast analog computing with emergent transient states) project was started by the European Union (EU) to develop AMS neuromorphic chips; The SyNAPSE program sponsored by DARPA was started to support development of neuromorphic chips by IBM, HRL Labs and other collaborators. Henry Markram started the Blue brain project at EPFL, Switzerland, which aims to simulate large-scale biologically-accurate neural networks on the IBM Blue Gene/L supercomputer.

4) In 2008, HP Labs developed the first memristor prototype, and demonstrated the first hybrid silicon CMOS/memristor chip. Memristors could be used as a type of basic elements to build neuromorphic chips.

5) In 2011, the BrainScaleS project, sponsored by EU, was started, which was aimed to develop large-scale neuromorphic supercomputers based on AMS circuits.

6) In 2012, the Blue brain project successfully simulated mesocircuits, each roughly consisting of 1 million neurons and 1 billion synapses, which was about the same scale as that of a honey bee brain. The simulations were 300 times slower than real time.

7) In 2013, the Human brain project (HBP), spondored by the EU, was started. HBP consists of 6 platforms, including: the Neuroinformatics platform, the Brain simulation platform, the High performance analytic and computing platform, the Medical informatics platform, the Neuromorphic computing platform, and the Neurorobotics platform.

8) In 2014, the SyNAPSE program, led by Dharmendra Modha, delivered the TrueNorth chip. Fabricated with 5.4 billion transistors, TrueNorth was IBM's largest chip to date. The power consumption was 70mW, 5000 times lower than a traditional processor with a similar amount of transistors. The chip was used in a visual saliency model with 3 million neurons[8], and it was capable of processing a 30fps video stream in real time, with only 200mW power consumption.

III. Neuron Models

When a neuron in the SNN receives an input spike, its membrane potential either increases or decreases, depending on whether the input spike is from an excitatory or an inhibitory synapse. If its membrane potential reaches the firing threshold, the neuron generates an outgoing spike, which travels down the axon to downstream synapses and neurons. When a spike arrives at a synapse, the pre-synaptic neuron releases neurotransmitters into the synaptic cleft, which in turn bind to dendritic receptors of the post-synaptic neuron, causing a change in the membrane potential of the post-synaptic neuron. Such behavior dynamics can be modeled at different levels of abstraction, ranging from Hodgkin-Huxley (HH) model, which is the most biologically realistic, to the Leaky integrate-and-fire (LIF) model, which is the

simplest and most computationally efficient, and many other models in between. As shown in Fig.1[9], the computational cost of a spiking neuron model grows with increasing biological accuracy.

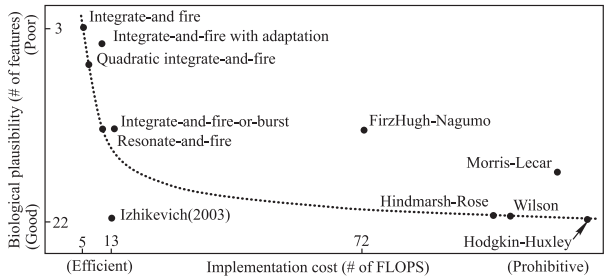

Fig. 1. Spiking neuron models at different levels of abstraction and their computational cost[9]. "# of FLOPS" is an approximate number of floating point operations (addition, multiplication, *etc.*) needed to simulate the model during a 1ms time span

1. Hodgkin-Huxley model

In HH model, neural membrane is modeled as an equivalent electrical circuit, as shown in Fig.2[10]. The lipid bilayer of the membrane is modeled as a capacitor, with C denoting the capacitance. The membrane potential V is the voltage across the capacitor. The ion channels on the membrane, including those for sodium, potassium, and leakage current, are modeled as resistors, with R_{Na}, R_{K}, and R_{le} denoting the resistances, respectively. R_{Na} and R_{K} may vary with time, whereas R_{le} is a constant. For each type of ions, due to the difference in concentration of the ions across the membrane, there exists an equilibrium potential. If the membrane potential is equal to the equilibrium potential of an ion type, the electric current, caused by movement of ions of this type, is zero. The equilibrium potential for sodium, potassium, and the leakage current are denoted by E_{Na}, E_{K}, and E_{le}, respectively. When the membrane potential V rises to a certain level, the neuron fires an output spike (also called an action potential), which is transmitted to downstream neurons, and the membrane potential is immediately reset. After firing an output spike, the neuron enters the refractory period, during which the neuron is not responsive to any input spikes.

The HH model can be either point neuron models or more detailed multi-compartment neuron models. In multi-compartment neuron models, different parts (*i.e.*, compartments) of the membrane may have different physioelectrical characteristics. In this case, each compartment can be modeled with a separate set of differential equations with different parameter values. The HH model can precisely reproduce experimental data observed in neurophysiological experiments, provided that the parameters are set properly. However, the HH model is computationally expensive, and simulations are typically hundreds of times slower than real-time, even with powerful supercomputers as in the Blue brain project.

2. Leaky integrate-and-fire model

There are several variants of LIF model. The simplest and the most widely used variant is the current-based LIF model, which is much more computationally efficient than the HH model, but less biologically accurate. The membrane potential V of a LIF neuron is governed by the following equations[10]:

$$C_{\mathrm{m}}\frac{\mathrm{d}V}{\mathrm{d}t} = g_{\mathrm{le}}(E_{\mathrm{le}} - V) + I$$
$$V = V_{\mathrm{reset}},\ \text{if } V \geq V_{\mathrm{th}}$$

where C_{m} denotes membrane capacitance, g_{le} denotes conductance (inverse of resistance) of the leakage channels, E_{le} denotes the equilibrium potential of the leakage channels, I denotes total input current.

Compared with the HH model, the LIF model replaces the complex differential equations for the conductances $g_{\mathrm{Na}}(t)$ and $g_{\mathrm{K}}(t)$ with a spike firing condition based on the firing threshold V_{th}.

Fig.3 and Fig.4[11] compare the dynamics of membrane potentials of an HH neuron and an LIF neuron with six input spikes. Input spikes are denoted by vertical dash-dotted lines in gray. The firing threshold V_{th} is denoted by the horizontal dashed line. Let's first consider an HH neuron in Fig.3.

Every time the neuron receives an input spike, a current is injected, causing a sudden increase in its membrane potential V. which is called the Excitatory postsynaptic potential (EPSP). If the time interval between two consecutive input spikes is long, *e.g.*, the time interval from 1ms to 23ms, the membrane potential decreases gradually due to the leakage current. Without subsequent input spikes, the membrane potential would eventually decrease to the equilibrium potential E_{le}. If enough input spikes arrive within a short time interval, *e.g.*, the three spikes arriving during the time interval from 44ms to 50ms, the membrane potential rises rapidly. When it is high enough to reach a threshold, the neuron fires an output spike. Immediately afterwards, the membrane potential is reset to a reset potential V_{reset} that is lower than the equilibrium potential E_{le}. Then the membrane potential gradually rises back to the equilibrium potential E_{le}. (Note that the numbers shown in the figures are not biologically realistic. For instance, the equilibrium potential of biological neurons is typically –70mV instead of 0mV). The LIF neuron in Fig.4 exhibits similar behavior as the HH neuron in Fig.3, with simplified dynamics, *e.g.*, the membrane potential rises instantaneously upon each input spike, and it is reset to the equilibrium potential E_{le} instead of the reset potential V_{reset}. Both models capture the essential property of spiking neurons that the response of a neuron is closely related to the temporal structure of the input. A neuron acts like a coincidence detector:

Enough spikes arriving at the neuron within a short time interval can elicit an output spike from the neuron. However, an equal number of spikes scattered across a long time interval may not be able to trigger any spike from the neuron due to the leakage current. This property of spiking neurons can be used to implement pattern recognition based on temporally coded spike trains, which is vital to biological sensory systems.

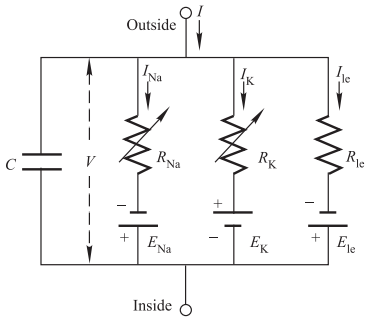

Fig. 2. Equivalent electrical circuit of the Hodgkin-Huxley model[10]

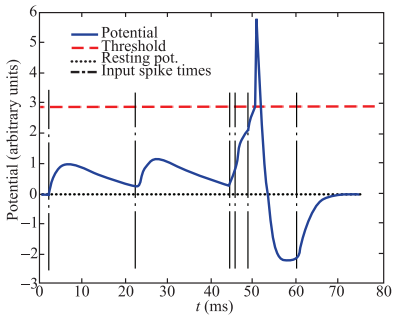

Fig. 3. Membrane potential of an HH neuron with input spikes

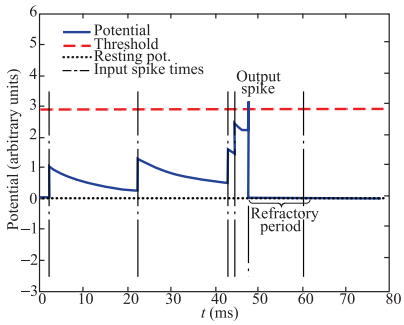

Fig. 4. Membrane potential of an LIF neuron with input spikes

3. Adaptive exponential integrate-and-fire (AdExIF) model

The AdExIF model[12] lies between the HH model and the LIF model in terms of biological accuracy and computational cost. The rate of increase of its membrane potential is slowed down every time a spike is fired, called Spike-triggered adaptation. As a consequence, the firing rate of the neuron gradually decreases under constant current injection. We can informally think of the neuron as getting "tired" as more and more spikes are fired. This behavior is consistent with the HH model.

4. Izhikevich model

The Izhikevich model[9] is close to the HH model in biological accuracy, yet its computational cost is close to that of the LIF or AdExIF model. The membrane potential of an Izhikevich neuron is governed by the following equations:

$$\begin{aligned} &\mathrm{d}V/\mathrm{d}t = 0.04V^2 + 5V + 140 - U + I \\ &\mathrm{d}U/\mathrm{d}t = a(bV - U) \\ &\{V = c,\ U = U + d\},\ \text{if } V \geq 30 \end{aligned}$$

where U is an auxiliary variable. With proper choice of parameters a, b, c, and d, the Izhikevich model can closely mimic the behavior of the HH model.

The choice of modeling abstraction levels should be made to serve the purpose of the simulation. If the goal is to provide efficient simulation tools for neuroscience research, then more biologically-realistic neuron models, *e.g.*, the HH model, should be used, with its high cost in terms of computation power. But if the goal is to develop information processing algorithms and chips based on SNN, then simpler neuron models, *e.g.*, LIF, AdExIF or Izhikevich, are sufficient for the purpose, and additional complexities brought by more complex models, *e.g.*, the HH model, are mostly not helpful to improve algorithm performance.

It should be noted that biological neural networks are far more complex than even the most detailed multi-compartment HH model[13], illustrated below (among many others):

1) Dendritic spikes. A dendrite can generate a spike and transmit it to the soma by itself. Depending on the location and time of its occurrence, a dendritic spike may either trigger or suppress a full spike at the soma. The average number of synapses per neuron is around 10,000. A neuron with thousands of synapses has powerful learning capability similar to a full neural network with point neurons.

2) Glia cells. Glia cells have been considered to provide support and protection for neurons, and they are ignored by most models in neuromorphic computing. Recently, neuroscientists found that glia cells may play a key role in information processing and communication, hence it cannot be ignored for accurate modeling.

3) Gene regulation triggered by neurotransmitters. Although most cells in an organism have identical DNA, neurons typically have a genome that is different from that of other cells. Neurons may dynamically alter their genome to satisfy demands of different information processing tasks, which facilitates self-adaption.

4) Electrical synapses. Apart from synapses based on chemical neurotransmitters, some synapses carry out communication via electrical signals.

IV. SNN Training Algorithms

Training algorithms for SNNs can be divided into several categories:

1) Unsupervised learning based on Hebb's rule, stated informally as: "Neurons that fire together, wire together." More formally, an increase in synaptic efficacy arises from the pre-synaptic neuron's repeated and persistent

stimulation of the post-synaptic neuron. Spiking-timing-dependent plasticity (STDP) is a Hebbian learning rule for SNNs: for two connected neurons, if the pre-synaptic neuron A always fires within a small time window before the post-synaptic neuron B fires, which means firing of B is correlated with firing of A, then the strength of the synapse between A and B is increased; on the contrary, if the pre-synaptic neuron A always fires within a small time window after the post-synaptic neuron B fires, the strength of the synapse between them is decreased. STDP was discovered from observations made in neurophysiological experiments.

2) Supervised learning. Like ANN training, many supervised learning algorithms for SNNs have been proposed, including: SpikeProp[14], Tempotron[15], Remote supervised method (ReSuMe)[16], Chronotron[17], Spike pattern association neuron (SPAN)[18], Precise-spike-driven (PSD) synaptic plasticity[19], *etc*. In contrast to STDP, these supervised training algorithms are designed by engineers, and are not based on biology.

3) Reinforcement learning. Via interaction with the environment, parameters that results in good performance is chosen based on reward and punishment.

4) Evolutionary Algorithms. A population of individuals (solutions) are maintained. Optimal configurations of parameters are chosen based on principles of biological evolution (cross-over, mutation, and natural selection).

Compared with ANN training algorithms, SNN training algorithms are still immature. Some researchers propose to convert traditional ANNs to SNNs[20]. A traditional ANN is first trained using a mature supervised or unsupervised learning algorithm, then converted to an SNN, thus avoiding the difficulty of training an SNN directly. Cao *et al.*[21] converted a Convolutional neural network (CNN) to a spiking CNN, which achieved accuracy similar to that of the original CNN on object recognition benchmarks Neovision2 and CIFAR-10. Neil *et al.*[22] converted a Deep belief network (DBN) to a spiking DBN, which achieved accuracy similar to that of the original DBN on the MNIST benchmark.

Liquid state machine (LSM)[23] is another SNN architecture that avoids training SNNs directly. A LSM is similar to a ANN-based Echo state machine (ESM), a recurrent neural network where connections between neurons and their weights are randomly generated and kept fixed. The LSM acts like a "reservoir", which maps external input to a high-dimensional space, making it easier for classification. The edge weights in a LSM are initialized with random values and kept fixed; a downstream classifier, which takes the state of the LSM as input and produces a classification result, is trained with conventional supervised learning algorithms. The learning capacity of LSM increases with its complexity. Due to the recurrent connections, LSM is capable of memorizing past information, which allows it to process inputs with a time dimension, such as audio or video.

V. Major Projects and Achievements

Table 1[24−32] summarizes major research projects in neuromorphic computing. We consider the following aspects: ① Implementation technology: multicore processors, digital circuits, or AMS circuits; ② Neuron model: LIF, AdExIF, Izhikevich, Quadratic integrate-and-fire (QIF), *etc*; ③ Training algorithm: whether on-chip learning, *i.e.*, dynamic tuning of synaptic strength, is supported or not; ④ The maximum SNN size supported, including number of neurons and synapses.

We made a few observations:

1) Most systems are based on hardware implementation, either digital circuits or AMS circuits, except SpiNNaker, which is based on software running on multicore ARM processors. AMS circuit implementations take advantage of physical characteristics of analog circuits to emulate the continuous neural dynamics directly, achieving much better computation and power efficiency than digital implementations. Especially, sub-threshold AMS circuits are able to achieve ultra-low power consumption.

2) Some systems execute faster than real time, *e.g.*, the HICANN chip from University of Heidelberg is 10,000 times faster than real time. Wafer-scale integration, *i.e.*, multiple chips are integrated on an uncut wafer, is employed to achieve massive parallelism. The scientific goal of HICANN is to provide neuroscientists with supercomputers to accelerate large-scale SNN simulations, rather than develop low-power intelligent embedded systems, which requires real-time operation.

3) Some systems, such as IBM TrueNorth, do not support on-chip learning, *i.e.*, the SNN parameters, including both the topology of the SNN and weights of the connections between the neurons, are fixed after deployment. The training is done offline on a powerful CPU/GPU-based platform, then the chip is configured with the trained SNN parameters. Without on-chip training, the design can be dramatically simplified. However, these chips are not able to self-adapt; any change of the SNN parameters requires reconfiguration and reboot.

4) There are a few related projects in China, including the Tianji chip from Tsinghua University[33], and the Darwin chip from Zhejiang University[34,35], but they are still in a preliminary stage compared to the large-scale projects listed in Table 1.

VI. Applications in Brain-Computer Interfaces

Brain-computer interface (BCI) technology aims to establish communication paths between biological brains or

Table 1. Major neuromorphic computing projects

Project / Organization	Implementation technology	Neuron model	Learning algorithm	# Neurons	# Synapses
SpiNNaker[24] / Univ. Manchester, UK	18-core ARM chip, connected by Network-on-Chip	LIF, Izhikevich	STDP	Up to 1K neurons per ARM core, up to 2.5M cores	1K synapses per neuron
TrueNorth[25] / IBM, USA	Digital neurons	LIF	None	256 neurons per neurosynaptic core; 4,096 cores per chip	1K synapses per neuron
HRL Labs[26], USA	Digital neurons, memristor synapses	Izhikevich	STDP	576 neurons per chip	70K time-multiplexed virtual synapses per chip
HICANN[27] / Univ. Heidelberg, Germany	AMS, wafer-scale integration	AdExIF	STDP	512 neurons per HICANN chip; 448 chips per wafer	224 synapses per neuron (in average)
Neurogrid[28] / Stanford Univ., USA	AMS (sub-threshold)	QIF	None	65K neurons per neural core; 16 neural cores per chip	375M synapses per chip
ETHZ[29], Switzerland	AMS (sub-threshold)	AdExIF	STDP	32 neurons per chip	32 synapses per neuron
BlueHive[30] / Cambridge Univ., UK	Digital neurons, multi-FPGA cluster (16 boards with four FPGAs each)	Izhikevich	None	65K neurons per FPGA	1K synapses per neuron
EMBRACE[31] / Univ. Ulster, National Univ., Ireland	AMS, connected by hierarchical Network-on-Chip	LIF	Genetic algorithms	32 (16 input+16 output) neurons per Modular neural time (MNT)	144 synapses per input neuron, 17 synapses per output neuron
IFAT[32] / UCSD, USA	AMS	Two-compartment LIF	None	65K neurons per chip	N/A

cultured neurons and external devices. A closed-loop BCI is a bi-directional communication mechanism between a biological brain and a computer for both perception and control. Researchers have developed implantable wireless sensors for collecting neural signals, *e.g.*, Neural Dust[36] from UC Berkeley, but they are typically only capable of data collection and transmission, not data processing. The data are transmitted wirelessly to an external workstation for processing. If we can implant an ultra low-power chip with on-chip or separate sensors, then we can build a closed-loop BCI system without replying on external workstation or wireless network connection. Many research challenges need to be overcome to achieve this vision: The chip must be compatible with biological tissue, which means that it needs to have ultra-low power consumption and heat dissipation. The chip must be capable of continuous operation without physical battery replacement, either through wireless changing, or through energy-harvesting. The chip must have adequate performance to carry out on-chip computation tasks, *e.g.*, real-time decoding of spike trains. The chip may be neuromorphic or not.

We summarize several preliminary research results of using neuromorphic algorithms or chips to perform neural decoding or control:

1) Vogelstein *et al.* developed SiCPG, a low-power chip based on AMS circuits[37]. A SNN model with 10 LIF neurons and 190 synapses was run on the chip to simulate the Central pattern generator (CPG) in vertebates. The chip was used to control a hind limb locomotion in a functionally paralyzed cat, so the cat could walk by itself. A long-term goal of the project was to enable patients who are paralyzed due to spinal column injuries to walk at their own pace.

2) Dethier *et al.* developed a SNN model for decoding neural signals from the motor cortex in a monkey[38]. The SNN consisted of 2,000 LIF neurons and was implemented using the Neural engineering framework (NEF)[39]. The SNN model was used in place of a Kalman-filter-based decoder in a BCI-based neural prosthesis, which used the neural signals from the motor cortex to control a robotic arm.

3) Ghaderi *et al.* developed an ultra-low-power neural signal processing chip based on analog circuits[40]. The chip was used in a cognitive prosthesis for rat hippocampus, which helped restore memory function. The chip was based on traditional signal processing algorithms for function approximation. It may be possible to use an implantable neuromorphic chip to simulate missing or malfunctioning functions of a brain region, interacting closely with other brain regions to restore certain damaged brain functions.

VII. Neuromorphic Sensors

Apart from computing systems based on principles of neuromorphic computing, there are also neuromorphic sensors for vision and audition.

Traditional video cameras are frame-based, *i.e.*, the camera samples an image frame periodically according to a given frame rate and outputs a video stream that consists of a series of frames. A high frame rate leads to high video quality, but also demands a higher bandwidth to transmit the video stream. Neuromorphic video cameras, *e.g.*, Dynamic vision sensors (DVS)[41], work in an event-triggered manner by detecting changes in pixel intensity, inspired by the biological retina. When the change in the

intensity of a pixel exceeds a given threshold, either from dark to bright or from bright to dark, a spike is generated for the pixel. If there is no change in intensity of a pixel, then no spike is emitted for that pixel. Spike encoding is based on the Address event representation, where each spike is represented by a tuple of timestamp of generation and pixel address. Thanks to the event-triggered nature, DVS cameras can achieve very high temporal resolution (in the range of microseconds) with very low bit-rate compared to traditional high-speed cameras, which makes it very suitable for use in resource-constrained scenarios such as Unmanned arial vehicles to track high-speed moving objects. Neuromorphic auditory sensors[41] are based on similar design principles.

VIII. Resources

There are many open-source software tools for SNN modeling and simulation (see footnote), such as NEST[*1], Brian[*2] and NEURON[*3]. PyNN[*4], developed at University of Heidelberg, is a SNN modeling language, which can be used as a frontend for software simulators such as NEST, Brian and NEURON, or neuromorphic chips such as HICANN and SpiNNaker. CARLsim[*5] is a GPU-based SNN simulator; Nengo[*6] is a SNN simulator based on Neural engineering framework (NEF). Spaun, "the world's largest functional brain model", is built based on Nengo, and is able to reproduce high-level cognitive function, such as vision, memory and inference.

IX. Conclusions

Neuromorphic computing is a promising research topic that is fraught with challenges, a few of which are listed below:

Training algorithm Compared with training algorithms for ANNs, training algorithms for SNNs are not as mature, especially in that the training algorithms are not able to effectively train a deep SNN with multiple hidden layers. Performance (classification accuracies) of SNNs on benchmark datasets commonly used in the industry, such as MNIST for handwritten digits recognition, is usually worse than that of ANN-based deep neural networks.

Low-power computation A main advantage of neuromorphic computing is low power consumption. However, ANN hardware accelerators based on digital circuits, such as the many CNN accelerators both in industry and academia, *e.g.*, the NeuFlow co-processor[42], the Diannao series[43], and commercial products from Cadence, Synopsys and others, can also achieve low power consumption by innovations in the architecture.

Programming model Although sub-threshold AMS chips can achieve ultra-low power consumption, applications based on such chips are very difficult to develop. Compared with digital circuits, AMS circuits are much more difficult to program, since physical characteristics of analog circuits have to be taken into account during design.

Currently, Neuromorphic computing has not gained wide industry acceptance, compared to deep learning systems based on ANN. However, we expect it to gain increasing importance in the near future, with the increasing demand for high performance low-power computing systems for intelligent information processing.

References

[1] H. Esmaeilzadeh, E. Blem, R.S. Amant, *et al.*, "Power challenges may end the multicore era", *Commun. ACM*, Vol.56, No.2, pp.93–102, 2013.

[2] R. Colwell, "The chip design game at the end of Moores law", *IEEE Hot Chips 25 Symposium* (*HCS*), pp.1–16, 1990.

[3] K. Grace, "Brain performance in TEPS", *http://aiimpacts.org/brain-performance-in-teps/*, 2015-5-6.

[4] Sze V, Chen Y H, Yang T J, *et al.*, "Efficient processing of deep neural networks: A tutorial and survey", arXiv preprint arXiv:1703.09039, *https://arxiv.org/abs/1703.09039*, 2017.

[5] Z. Wu, Y. Zhou, Z. Shi, *et al.*, "Cyborg intelligence: Recent progresses and future directions", IEEE Intelligent Systems, Vol.31, No.6, pp.44–50, 2016.

[6] Y. Wang, M. Lu, Z. Wu, *et al.*, "Visual cue-guided rat cyborg for automatic navigation", IEEE Computational Intelligence Magazine, Vol.10, No.2, pp.42-52, 2015.

[7] D. Ma, J. Shen, Z. Gu, *et al.*, "Darwin: A neuromorphic hardware co-processor based on spiking neural networks", Journal of Systems Architecture - Embedded Systems Design, Vol.77, pp.43–51, 2017.

[8] A. Andreopoulos, B. Taba, A.S. Cassidy, *et al.*, "Visual saliency on networks of neurosynaptic cores", *IBM J. Res. Dev.*, Vol.59, No.2/3, pp.1–9, 2015.

[9] E.M. Izhikevich, "Which model to use for cortical spiking neurons?", *IEEE Trans. Neural Networks*, Vol.15, No.5, pp.1063–1070, 2004.

[10] P. Dayan and L.F. Abbott, *Theoretical Neuroscience*, MIT Press, Cambridge, MA, USA, 2001.

[11] T. Masquelier, R. Guyonneau and S.J. Thorpe, "Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains", *PLoS One*, Vol.3, No.1, doi:10.1371/journal.pone.0001377, 2008.

[12] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity", *J. Neurophysiol.*, Vol.94, No.5, pp.3637–3642, 2005.

[13] T. Dettmers, "The Brain vs Deep Learning", *http://timdettmers.com/2015/07/27/brain-vs-deep-learning-singularity/*, 2015-7-27.

[14] S.M. Bohte, J.N. Kok and H. La Poutré, "SpikeProp: Backpropagation for networks of spiking neurons", *Proc. of 8th European Symposium on Artificial Neural Networks*, pp.419–425, 2000.

[*1] *http://www.nest-initiative.org/*
[*2] *http://briansimulator.org/*
[*3] *http://www.neuron.yale.edu/*
[*4] *http://www.neuralensemble.org/PyNN/*
[*5] *http://www.socsci.uci.edu/~jkrichma/CARL/*
[*6] *http://www.nengo.ca/*

[15] R. Gütig and H. Sompolinsky, "The tempotron: A neuron that learns spike timing-based decisions", *Nat. Neurosci.*, Vol.9, No.3, pp.420–428, 2006.

[16] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting", *Neural Comput.*, Vol.22, No.2, pp.467–510, 2010.

[17] R.V. Florian, "The chronotron: A neuron that learns to fire temporally precise spike patterns", *PLoS One*, Vol.7, No.8, doi:10.1371/journal.pone.0040233, 2012.

[18] A. Mohemmed, S. Schliebs, S. Matsuda, *et al.*, "Span: Spike pattern association neuron for learning spatio-temporal spike patterns", *Int. J. Neural Syst.*, Vol.22, No.04, Artical ID 1250012, 17 pages, 2012.

[19] Q. Yu, H. Tang, K.C. Tan, *et al.*, "Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns", *PLoS One*, Vol.8, No.11, doi:10.1371/journal.pone.0078318, 2013.

[20] P.U. Diehl, D. Neil, J. Binas, *et al.*, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing", *International Joint Conference on Neural Networks* (*IJCNN*), pp.1–8, 2015.

[21] Y. Cao, Y. Chen and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition", *Int. J. Comput. Vis.*, Vol.113, No.1, pp.54–66, 2015.

[22] D. Neil and S.-C. Liu, "Minitaur, an event-driven FPGA-based spiking network accelerator", *IEEE Trans. Very Large Scale Integr. Syst.*, Vol.22, No.12, pp.2621–2628, 2014.

[23] W. Maass, T. Natschläger and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations", *Neural Comput.*, Vol.14, No.11, pp.2531–2560, 2002.

[24] S.B. Furber, F. Galluppi, S. Temple, *et al.*, "The spinnaker project", *Proc. IEEE*, Vol.102, No.5, pp.652–665, 2014.

[25] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface", *Science*, Vol.345, No.6197, pp.668–673, 2014.

[26] J.M. Cruz-Albrecht, T. Derosier and N. Srinivasa, "A scalable neural chip with synaptic electronics using CMOS integrated memristors", *Nanotechnology*, Vol.24, No.38, Artical ID 384011, 11 pages, 2013.

[27] J. Schemmel, D. Briiderle, A. Griibl, *et al.*, "A wafer-scale neuromorphic hardware system for large-scale neural modeling", *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (*ISCAS*), pp.1947–1950, 2010.

[28] B.V. Benjamin, P. Gao, E. McQuinn, *et al.*, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations", *Proc. IEEE*, Vol.102, No.5, pp.699–716, 2014.

[29] S. Moradi and G. Indiveri, "An event-based neural network architecture with an asynchronous programmable synaptic memory", *IEEE Trans. Biomed. Circuits Syst.*, Vol.8, No.1, pp.98–107, 2014.

[30] S.W. Moore, P.J. Fox, S.J.T. Marsh, *et al.*, "Bluehive – A field-programable custom computing machine for extreme-scale real-time neural network simulation", *IEEE 20th Annual International Symposium on Field-Programmable Custom Computing Machines* (*FCCM*), pp.133–140, 2012.

[31] S. Pande, F. Morgan, G. Smit, *et al.*, "Fixed latency on-chip interconnect for hardware spiking neural network architectures", *Parallel Comput.*, Vol.39, No.9, pp.357–371, 2013.

[32] J. Park, S. Ha, T. Yu, *et al.*, "A 65k-neuron 73-Mevents/s 22-pJ/event asynchronous micro-pipelined integrate-and-fire array transceiver", *IEEE Biomedical Circuits and Systems Conference* (*BioCAS*), pp.675–678, 2014.

[33] L. Shi, J. Pei, N. Deng, *et al.*, "Development of a neuromorphic computing system", *IEEE International Electron Devices Meeting* (*IEDM*), Washington, DC, USA, pp.4.3.1–4.3.4, 2015.

[34] J. Shen, D. Ma, Z. Gu, *et al.*, "Darwin: A neuromorphic hardware co-processor based on spiking neural networks", *Science China Information Sciences*, Vol.59, No.2, pp.1–5, 2016.

[35] D. Ma, J. Shen, Z. Gu, *et al.*, "Darwin: A neuromorphic hardware co-processor based on spiking neural networks", *Journal of Systems Architecture*, Vol.77, pp.43–51, 2017.

[36] D. Seo, J.M. Carmena, J.M. Rabaey, *et al.*, "Neural dust: An ultrasonic, low power solution for chronic brain-machine interfaces", arXiv preprint arXiv:1307.2196, 2013.

[37] R.J. Vogelstein, F.V.G. Tenore, L. Guevremont, *et al.*, "A silicon central pattern generator controls locomotion in vivo", *IEEE Trans. Biomed. Circuits Syst.*, Vol.2, No.3, pp.212–222, 2008.

[38] J. Dethier, P. Nuyujukian, S.I. Ryu, *et al.*, "Design and validation of a real-time spiking-neural-network decoder for brain-machine interfaces", *J. Neural Eng.*, Vol.10, No.3, Artical ID 036008, 25 pages, 2013.

[39] C. Eliasmith and C.H. Anderson, *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*, MIT Press, London, England, 2004.

[40] V.S. Ghaderi, D. Song, J. Choma, *et al.*, "Nonlinear cognitive signal processing in ultralow-power programmable analog hardware", *IEEE Trans. Circuits Syst. II Express Briefs*, Vol.62, No.2, pp.124–128, 2015.

[41] S.-C. Liu and T. Delbruck, "Neuromorphic sensory systems", *Curr. Opin. Neurobiol.*, Vol.20, No.3, pp.288–295, 2010.

[42] C. Farabet, B. Martini, B. Corda, *et al.*, "Neuflow: A runtime reconfigurable dataflow processor for vision", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (*CVPRW*), pp.109–116, 2011.

[43] Du Z, Liu S, Fasthuber R, *et al.*, "An accelerator for high efficient vision processing". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.36, No.2, pp.227–240, 2017.

ZHANG Ming received the B.S. degree in software engineering from Zhejiang University in 2010, and is currently a Ph.D. student at Zhejiang University. (Email: editing@zju.edu.cn)

GU Zonghua (corresponding author) received the Ph.D. degree in computer science and engineering from the University of Michigan at Ann Arbor in 2004. He is currently an associate professor in the College of Computer Science, Zhejiang University. (Email: zonghua@gmail.com)

PAN Gang received the B.S. and Ph.D. degrees in computer science in 1998 and 2004, respectively, from Zhejiang University. He is a professor of the College of Computer Science and Technology, Zhejiang University, China. (Email: gpan@zju.edu.cn)