TOPICAL REVIEW

Emerging memory technologies for neuromorphic computing

IOP Publishing Nanotechnology
Nanotechnology 30 (2019) 032001 (33pp) https://doi.org/10.1088/1361-6528/aae975

Topical Review

Emerging memory technologies for neuromorphic computing

Chul-Heung Kim[1], Suhwan Lim[1], Sung Yun Woo, Won-Mook Kang, Young-Tak Seo, Sung-Tae Lee, Soochang Lee, Dongseok Kwon, Seongbin Oh, Yoohyun Noh, Hyeongsu Kim, Jangsaeng Kim, Jong-Ho Bae and Jong-Ho Lee

The Department of Electrical and Computer Engineering and Inter-university Semiconductor Research Center (ISRC), Seoul National University, Seoul 08826, Republic of Korea

E-mail: jhl@snu.ac.kr

Received 9 January 2018, revised 21 August 2018
Accepted for publication 18 October 2018
Published 13 November 2018

Abstract

In this paper, we reviewed the recent trends on neuromorphic computing using emerging memory technologies. Two representative learning algorithms used to implement a hardware-based neural network are described as a bio-inspired learning algorithm and software-based learning algorithm, in particular back-propagation. The requirements of the synaptic device to apply each algorithm were analyzed. Then, we reviewed the research trends of synaptic devices to implement an artificial neural network.

Keywords: neuromorphic computing, emerging memory, spike-timing-dependent plasticity (STDP), spike-rate-dependent plasticity (SRDP), back-propagation (BP), synaptic device

(Some figures may appear in colour only in the online journal)

1. Introduction

Recently, machine learning has attracted a great deal of attention in the IT industry and is being developed rapidly with the performance enhancement of the graphics processing unit (GPU)-based hardware accelerator. Although there are a variety of algorithms in machine learning, deep neural network (DNN) technology based on the back-propagation (BP) algorithm has shown excellent performance in many areas including image, speech recognition, and translation, even sometimes outperforming human cognitive abilities [1–6]. The state-of-the-art architectures of DNNs include convolutional neural networks (CNNs) and recurrent neural networks (RNNs). However, there are important challenges with regard to power consumption, the occupied area of the hardware platform and training times. Therefore, the need for implementing neuromorphic artificial neural networks (ANNs) with low power and small area has been emerging [7, 8]. Table 1 summarizes different types of neuromorphic ANNs and their respective features. The human brain described in the left side of the table has the very powerful ability to recognize real-world problems with extremely low power, but the learning mechanism and structure of the human brain are not yet clearly defined. Deep learning shown on the right side is based on the software-based learning algorithms and the von Neumann computer architectures. It is also powerful in recognition tasks, but the power consumption is very high. In ANNs using neuromorphic technology, learning algorithms can be classified into two categories: bio-inspired learning algorithms and software-based learning algorithms [9–11]. The learning algorithms based on the bio-inspired approach, such as spike-timing-dependent plasticity (STDP) and spike-rate-dependent plasticity (SRDP), implement the model of the biological neuron cell behavior [12]. In the case of bio-inspired learning algorithms, there are two subcategories: supervised and unsupervised learning. Research to implement ANNs using bio-inspired learning algorithms is biased towards using unsupervised learning. However, since supervised learning can be efficient in certain

[1] These authors contributed equally to this work.

Table 1. Different types of neural networks and their respective features.

	Human brain	Neuromorphic		Deep learning
Target	Biology	Spiking neural networks (SNNs)	Deep neural networks (DNNs) Convolutional neural networks (CNNs) Recurrent neural networks (RNNs)	
		HW-based		SW-based
Components	Neuron array	Neuron array (Integrate & fire)	Neuron array (Activation function, Integrate & fire)	von Neumann architecture (GPU, TPU, etc.)
	Synapse array	Synaptic device array	Synaptic device array	
Learning algorithm	STDP, SRDP, etc.	STDP, SRDP (Bio-inspired)	Back-propagation (Software-based)	
Power consumption	Extremely low	Low	Intermediate	High
Maturity	Extremely high	Low	Intermediate	High

application fields, the above two learning methods are being studied in parallel. The neural networks based on bio-inspired algorithms have the advantage of low power, because the neural network is capable of event-driven operation and on-chip learning, similar to the biological brain. It is very important for memory technologies to implement ANN using these algorithms to have their own synapse weight-update scheme in order to learn by themselves without the help of external computation systems. In addition, the weight-update method and endurance are also significant for reducing the time and power consumption in the learning process. The retention characteristics of the weight itself should also be considered to prevent errors in the inference process in the learned synapse array. In contrast, the software-based learning algorithms, especially BP, are based on the mathematical model and optimize the hypothesis of the neural network by minimizing the training and generalization errors [13]. There are various approaches to accelerate the DNNs using the BP algorithm such as Google's tensor processing unit (TPU), MIT's Eyeriss, the University of Utah's ISAAC, CAS's DaDianNao and UCSB's PRIME [14–18]. Among them, the use of emerging nonvolatile memory (NVM) devices has been widely studied because they provide high scalability and enable high-speed parallel operations with extremely low power. Therefore, this approach using the emerging NVMs can also be called a DNN accelerator. Electronic synaptic devices can represent the weight values of neural networks with their multi-level conductance values, and perform massively parallel computations using these conductance values. Here, we focus on approaches to accelerate DNNs using electronic synaptic devices, and these approaches are called hardware-based deep neural networks (HW-DNNs). The HW-DNNs are divided into off-chip training and on-chip training. On-chip training can provide low power and high-speed learning, while off-chip training can be applied to more complex neural networks. It is very hard to represent software-level high-precision weight values as the analog values of actual electronic devices. These actual electronic devices have non-ideal characteristics such as nonlinearity of conductivity response, asymmetry, finite number of conductance values, limited endurance and variation of the NVM device itself. Therefore, in the study of accelerating DNNs using electronic synaptic devices, the non-ideal characteristics of these devices should be fully considered.

Furthermore, we also review the deep spiking neural networks (DSNNs) that combine the DNNs with spiking signal domains. Although there are not many studies to implement DSNNs using electronic synaptic devices, applying the low-power characteristics of electronic devices to event-driven systems of SNNs can be a promising field of neuromorphic research.

In this paper, recent trends in the implementation of neuromorphic computing using emerging memory technology are reviewed. Various devices such as resistive memory (RRAM), conductive-bridge memory (CBRAM), phase change memory (PCM), spin-based memory and field-effect transistor (FET)-based memory have been reported as a synaptic device. The requirements of the synaptic devices for implementing a low-power and highly integrated neural network are discussed. We present the conclusion by analyzing current research trends of emerging memory devices for neuromorphic computing and additional research challenges for successful neuromorphic computing chips. The goal of this work is to motivate more advanced research work by sharing the current global research status in neuromorphic computing.

2. Learning algorithms for implementing ANN

2.1. Bio-inspired learning algorithms (STDP/SRDP)

In this chapter, we introduce bio-inspired learning algorithms and the features of ANNs implemented using these

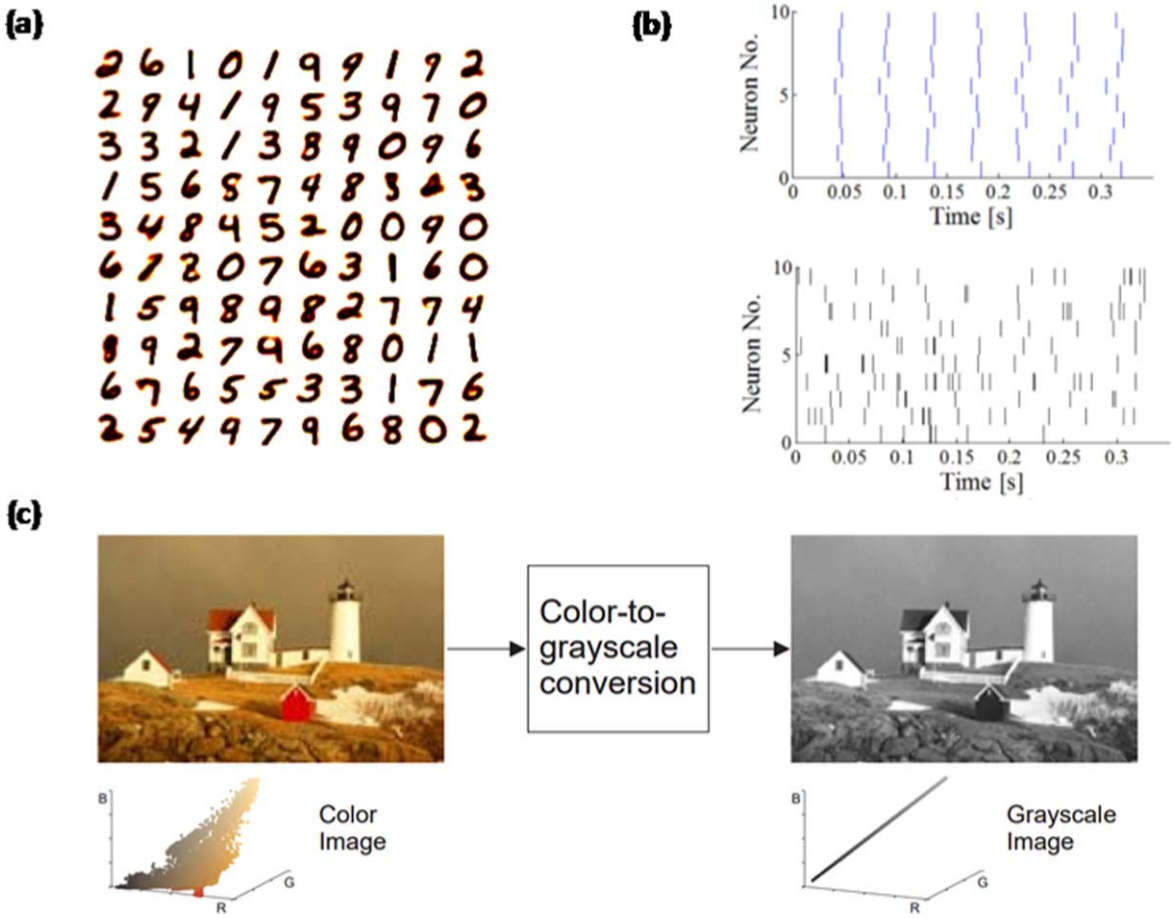

Figure 1. (a) Simple binary MNIST data set used to evaluate the performance of a network. CC BY 4.0. Reproduced from [20]. CC BY 4.0. (b) Scheme of input pulse train presented by (top) temporal encoding and (bottom) rate encoding. IEEE. © [2011] IEEE. Reprinted, with permission, from [36]. (c) Complex input data such as colorful images should be encoded in grayscale. [21] John Wiley & Sons. © 2008 The Author(s) Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

algorithms. There are two representative bio-inspired learning algorithms: STDP and SRDP. These algorithms are learning methods developed from the learning mechanisms observed in the biological brain. STDP is a learning mechanism in which the synapse weight is changed by the time difference between the signal from the presynaptic neuron and the signal from the postsynaptic neuron. SRDP, another learning algorithm, determines the weight change of the synapse by the frequency of the signal from the presynaptic neuron applied to the synapse. Here, we introduce the neural encoding methods and classify the ANNs using bio-inspired learning algorithms as supervised/unsupervised learning. We then discuss the requirements of the synaptic devices used to implement the ANNs applying bio-inspired learning algorithms.

2.1.1. Methods of neural encoding. In order to train neural networks and have the ability to classify data, it is essential to transform the data into proper form. The method of converting data into input pulse is determined by various system components such as learning algorithms, inference methods, types of neuron circuit models and so on. The simplest data expression is binary form in which data is transmitted by only '0' and '1'. This form is advantageous for easy tasks [19] such as recognizing simple patterns such as those in figure 1(a) [20]. This is because the memory and time required for learning as well as the burden on peripheral circuits are reduced. In addition, binary data make a significant difference between input signals, which can maximize the effect of the input data. However, more complex data, such as a colorful image (as shown in figure 1(c)) [21] or gas mixture [22], should be expressed in grayscale rather than binary form to represent more information of the data. For instance, CIFAR-10 and 100 image data sets with 60 000 different RGB images should be presented in grayscale. Many neural coding rules to encode the data as a stimulus in the neuron system have been studied in various models, such as 'population coding' [23] and 'sparse coding' [24]. However, among various neural coding schemes, 'rate coding' and 'temporal coding' are the schemes most widely studied since they are easy to implement in hardware systems [25].

Rate coding is theoretically based on Bienenstock, Cooper and Munro theory [26]. The larger the value of the data, the greater the frequency of spike firing. In particular, many groups, including Indiveri [27], O'connor [28], Diehl [20, 29] and Querlioz [30] modeled input pulses as Poisson-distributed spike trains. They generated stochastic Poisson input pulse whose firing rate is proportional to the intensity of the input pixel. This stochastic scheme showed robustness to random

noise spikes [30]. Some groups, on the other hand, generated constant-frequency spike train which controls the firing rates only with the inter-spike time duration [31]. The rate-encoding scheme has the disadvantage that it is not appropriate for the rapid change of input stimuli in fast time scale. However, it notably shows high robustness with noise [30, 32]. Temporal coding, where information is represented by spike timing, has also been applied to many neural networks. In the temporal coding scheme, the larger the quantity of the data, the earlier the spike pulse is generated. For example, 0101100 sequence encoded temporally is considered differently from 0001011, even though the firing rates of sequences are the same [33]. Kaneko *et al* transformed the input data to pulse time information in the time clock [34]. In addition, Sheik constructed a bio-plausible network combining a temporal coding scheme with a leaky integrate & fire (LIF) neuron model [35]. Figure 1(b) shows the pulse schemes for presenting input pulse in two different encoding methods [36].

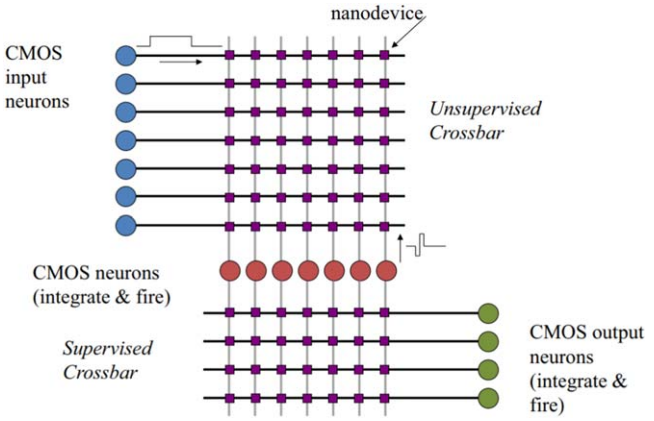

Figure 2. Topology of architecture combining an unsupervised and supervised layer. IEEE. © [2012] IEEE. Reprinted, with permission, from [39].

2.1.2. Supervised learning. Among the methods for implementing a hardware-based neural network (HNN), learning through supervision has largely been investigated in two ways: BP algorithm based on gradient descent and bio-inspired learning with teaching signals. In this chapter, we will only deal with the latter method. In bio-inspired learning, unsupervised learning is being studied more widely. However, conventional studies show several advantages in supervised learning and it is being studied with unsupervised learning. Kim *et al* explained that supervised learning has significantly higher performance over unsupervised learning with the same number of output neurons and synapses [37]. Although the modulation of synapse weight is controlled by the feedback spike from the integrate-and-fire (I&F) circuit in conventional unsupervised learning, this study used the feedback signal with exterior supervision. The teaching signal from the output layer is modeled as temporal coding to train synapses with an STDP algorithm. The target output neuron presents the teaching signal at late timing (compared to input signal) to potentiate the target synapses, and the other neurons present teaching signals at early timing. Yuan *et al* explained why the teaching signal is essential for supervised learning with synchronous input signals [38]. That is because the synchronously presented input cannot make the output spike precede the input signal. Hence, there will be only weight increment by the STDP rule, no weight decrement. Artificial stimulation is required to prevent the synapse weights from increasing excessively. Querlioz *et al* proposed another application of supervised learning for improving the performance of the network [30, 36, 39]. In a conventional network trained by unsupervised learning, the data represented by the neurons cannot be known because the data are trained only in the first firing random neuron. An additional labeling process is required to identify the results, and supervised learning can play this role in the next layer, as shown in figure 2. In other words, data are classified in the first layer trained without supervision and labeled in the second layer trained with supervision. As previously mentioned, this group also claims that supervised learning can reduce the number of neurons or synapses and achieve higher accuracy in simple methods [39]. On the other hand, networks trained by supervision can exhibit a low level of robustness in device variation, limiting the configuration and expansion of the synapse array. Supervised learning can be a burden on area and power because it requires peripheral circuits.

2.1.3. Unsupervised learning. In SNNs, complex cognitive computing, including online unsupervised learning and classification, has been effectively performed using bio-inspired learning algorithms [7, 9]. STDP, which is one of the most popular biologically plausible learning rules, has been exploited in an unsupervised fashion [40]. The local learning rule modulates synaptic weights between the presynaptic and postsynaptic neurons in accordance with the spike-timing difference [41]. The synaptic weight potentiates if a spiking of a postsynaptic neuron follows that of a presynaptic neuron, and vice versa. The smaller the spike-timing difference, the larger the synaptic weight variation. In an STDP-based SNN, a current through each synapse is fed into a postsynaptic neuron, and the neuron fires when its membrane potential exceeds the spiking threshold. Simultaneously, synaptic weights are updated, and lateral inhibition is commonly implemented in the form of winner-takes-all (WTA) architecture for competitive learning. Although a full computational architecture with the STDP local learning rule should be explored more, STDP-based unsupervised learning is efficient in distinguishing unlabeled or unstructured data and is advantageous for real-time data processing [42]. Online unsupervised learning performance based on the STDP local learning rule has been shown in system-level simulation works using memristive synapse and the LIF neuron model.

Diehl *et al* proposed a biologically plausible unsupervised learning mechanism including lateral inhibition and adaptive

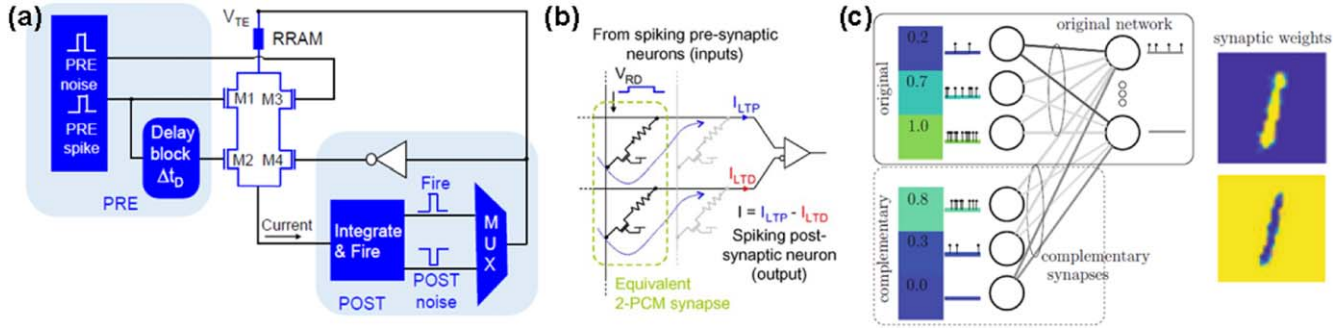

Figure 3. (a) Implementation of SRDP using a 4T1R synapse. Transistors M_1 and M_2 are associated with long-term potentiation, and long-term depression takes place via transistors M_3 and M_4. © [2016] IEEE. Reprinted, with permission, from [12]. (b) Schematic of a two-PCM synapse in a crossbar. © [2012] IEEE. Reprinted, with permission, from [45]. (c) Neuromorphic system based on an STDP algorithm using an encoding scheme with both original patterns and complementary patterns. [46] (2017) © Springer International Publishing AG 2017. With permission of Springer.

threshold [20]. In order to improve the biological plausibility of an SNN, the power law and exponential conductance-dependent synaptic plasticity were exploited, and the input pattern train was encoded to ensure a minimum firing rate of each postsynaptic neuron for competitive learning by increasing the maximum input firing rate. Using this SNN based on the STDP algorithm, a 95% classification performance on the MNIST handwritten data set was demonstrated with 6400 postsynaptic neurons in the two-layer system. Although remarkable classification results of unsupervised learning based on STDP have been executed, the demonstration requires additional circuitry for the fine-tuning of model parameters not suitable for processing various types of data.

For the straightforward implementation of an SNN using a nonvolatile memory array, Querlioz *et al* introduced a simplified STDP rule for selective pattern learning in an unsupervised manner [30, 36]. This simple learning rule is focused on the spiking of a postsynaptic neuron. When an output neuron fires, synapses contributed to the output spike are potentiated, while the other synapses connected to the output neuron are depressed. Thus, each output neuron is selectively specialized to different patterns. In [36], the simplified STDP scheme with a memristive device model was easily executed by overlapping pre- and postsynaptic spikes using simple pulse generation owing to its simplicity. In order to confirm the robustness of SNNs, the same group has investigated the impact of device variability including memristive synapse and CMOS neuron variability with system-level simulations in SNNs [30]. Realistic system-level simulation was performed by considering the actual variation caused by memristive synapse variability and sneak path issues in the crossbar array of two-terminal memristors. The system was immune to variations of synaptic devices up to 25%, and the homeostasis function effectively compensated for the significant reduction in the recognition rate by device variability. Immunity improvement for device variation is due to the proposed homeostasis of neurons. This biologically plausible property along with a WTA topology of lateral inhibition plays a key role in regulating the responsiveness of neurons equivalently, preventing ones with lower thresholds from firing predominantly in the network.

In addition to the previously introduced approaches for unsupervised learning, several different input encoding schemes, learning methods and system structures have been presented to enhance the performance of SNNs. Ambrogio *et al* proposed the input pulse scheme in which the input noise is exploited to depress background synapses [43]. During the unsupervised learning process in a fully connected two-layer system using the standard STDP learning rule, a pattern of MNIST digit and a random noise were presented alternatively. This configuration induced potentiation of pattern synapses and depression of background synapses for selective learning in an unsupervised fashion. In [44], specifically, the impact of input parameters associated with noise on pattern learning speed and efficiency was evaluated through both Monte Carlo and analytical models. This investigation can be useful to optimize input parameters for minimizing false firing by noise in the SNN using inherently nonlinear and asymmetric memristive devices. However, these works require additional circuitry for random noise generation, and it is hard to optimize input parameters of noise for processing various types of data.

Milo *et al* performed a simple learning task of an 8×8 pattern based on SRDP using a novel 4-transistor/1-resistor (4T1R) synapse [12]. Each synapse contains four transistors divided into two branches (figure 3(a)). Unlike STDP-based pattern learning, SRDP is implemented by controlling the degree of overlapping signals according to the frequency of the input signal when two identical signals are applied with a certain time difference. The biologically plausible SRDP learning rule was verified by simple learning simulations using a single output neuron.

To overcome the asymmetry issue of memristive devices used as synapses, Bichler *et al* designed a synapse with two-PCM devices so that the synaptic weight was encoded with the difference in conductivity between a pair of devices [45] (figure 3(b)). This configuration allows the implementation of negative synaptic weights. A conductance refresh mechanism

was introduced to avoid the saturation of device conductance while preserving the synaptic weight. With the simplified STDP learning rule, real-time car trajectory extraction of temporally correlated features from the dynamic vision sensor was performed in system-level simulation.

In contrast to most system-level simulation works, Sidler *et al* proposed the input encoding scheme using an additional complementary pattern and demonstrated online unsupervised pattern learning capabilities on the MNIST data set with both simulation and experimental results [46]. To apply this approach, the number of input neurons and synapses should be delivering information from a complementary pattern input, which is an inverted version of the original pattern input (figure 3(c)). Even though it requires additional area of the system, this configuration could be advantageous for the classification of overlapping features.

2.1.4. Requirements. To apply the bio-inspired learning algorithm, it is advantageous for the synaptic device to have two or three terminals, so that learning according to the time difference between pre- and postsynaptic signals or the frequency of presynaptic signals can be easily implemented. Generally, a specific pulse scheme is used for automatic synaptic weight update [47]. It is common practice to reduce the control of the peripheral circuits to a minimum for implementing a weight update in the synaptic device.

The devices used for bio-inspired learning include RRAM, CBRAM, PCM, spin-based memory and FET-based memory. In order to perform complex large-scale tasks, the basic requirement is the density of the device array. In general, most research groups are basically using crossbar arrays to construct large-scale parallel computing neural networks. Although two-terminal devices are attracting much attention because of their ease of implementation of crossbar arrays, in fact a two-terminal device requires a select device to eliminate the sneak path that occurs in a crossbar array configuration. Ultimately, in order to further increase the degree of integration, the form of a device capable of 3D integration is preferred [48]. Moreover, since the goal is not to implement a synaptic device-only array, but to implement a large-scale neural network system, the CMOS compatibility of the synaptic device is important. As a result, it is necessary to be compatible with CMOS technology for system implementation.

Energy efficiency in weight learning and inference processes in synaptic device arrays must also be carefully considered, and needs to be evaluated differently depending on the application. It is important to reduce the power used in the weight update in the case of a synaptic device array used in an application where continuous learning is to be performed in real time. In the case of a synaptic device array mainly used for the inference process, it is necessary to reduce the power at the weighted-sum operation.

In terms of the characteristics of a single synaptic device, analog memory characteristics should be examined [49]. The purpose of the neuromorphic synaptic array is to efficiently combine the multiplication results of the input signal with the weights of the memory devices having the analog weight. Therefore, having an analog memory characteristic is the most basic requirement of a memory device to be used as a synaptic device. As the number of weight levels increases, it is easy to perform complex tasks. Numerous studies have been made to realize gradual implementation of such analog memory characteristics [50]. In [50], Yu summarized the desirable performance metrics for synaptic devices, as shown in table 2. If it is difficult to achieve a gradual conductance change due to the inherent characteristics of the device, the gradual change may be implemented by controlling the pulse shape and adding additional devices (resistors or FETs) [47]. However, these additional devices make the overall circuit design complicated. In general, the conductance margin between the high-conductance state and low-conductance state of the device can have a significant impact on the performance of the neural network. This margin should be ensured to clearly distinguish the difference between the background signal and image signal when comparing weighted-sum values in the inference process. However, this is closely related to the size of the neural network depending on the application used. It should be noted that if the upper limit of the conductance value becomes too large, the power consumption of the entire system may increase sharply. Also, the aspect of change in the weight-update process is very important. It is usually described by nonlinearity and symmetry. In the course of implementing the gradual conductance, many research groups have mainly analyzed how the linearity of conductance change, symmetry characteristics of potentiation and depression process affect inference accuracy [51]. In [51], Kim *et al* reported that the nonlinearity in the conductance change of synaptic devices is not critical to the pattern recognition rate of the system, as shown in figure 4. In the case of using STDP and SRDP algorithms, which are widely used in bio-inspired learning, it is more important to prevent abrupt depression than to reinforce the linear potentiation characteristic of the device. In the case of supervised learning involving external

Table 2. Summary of the desirable performance metrics for synaptic devices. © [2018] IEEE. Reprinted, with permission, from [50].

Performance metrics	Desired targets
Device dimension	<10 nm
Multi-level states number	$>100^{*}$ (with linear and symmetric update)
Energy consumption	<10 fJ/programming pulse
Dynamic range (on/off ratio)	$>100^{*}$
Retention	>10 years* (for inference)
Endurance	$>10^{9}$ updates* (for online training)

Note: * these numbers are application dependent.

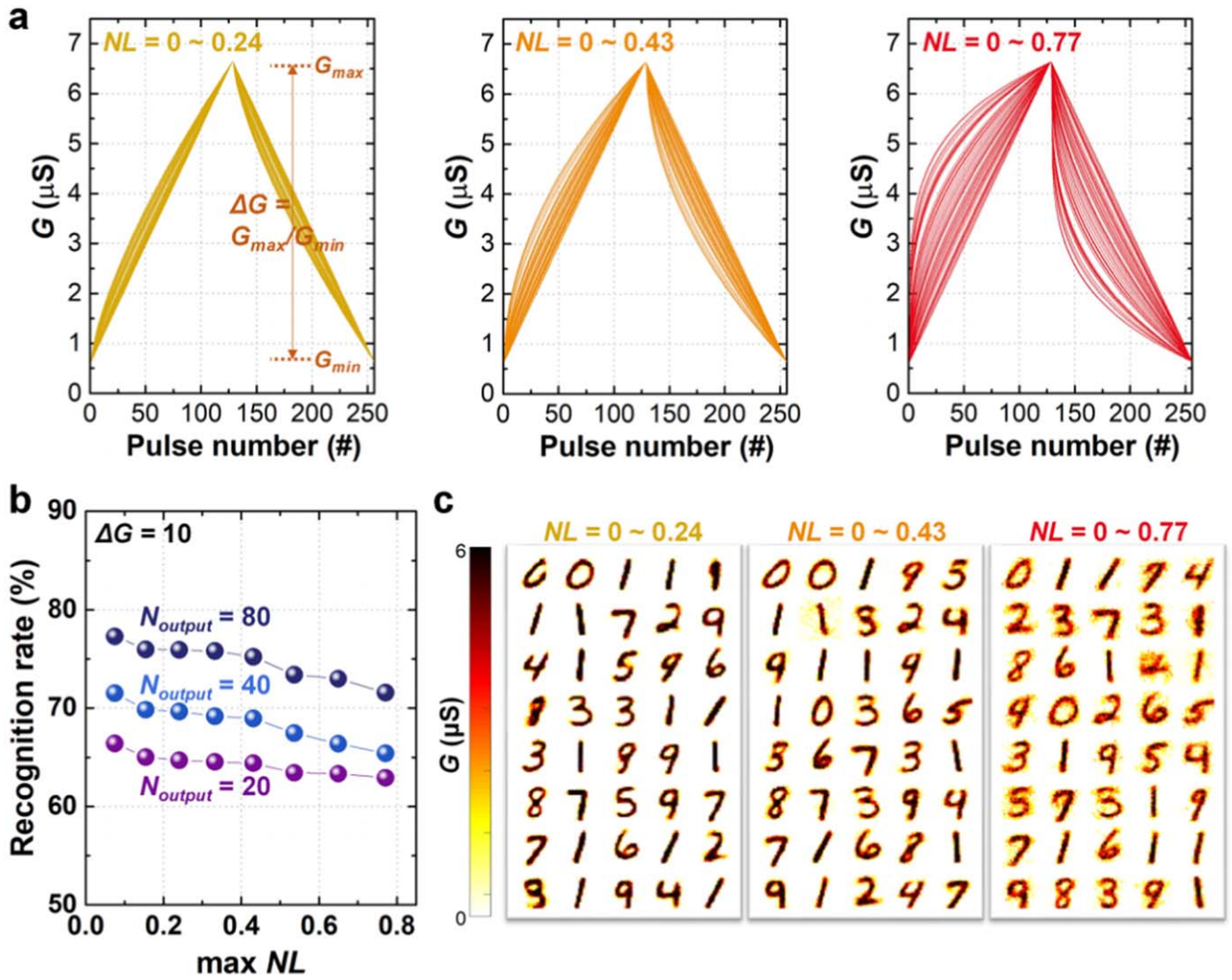

Figure 4. (a) Synaptic device conductance (G) as a function of applied pulse number with randomly assigned NL values. (b) Simulated recognition rate as a function of the maximum NL value after 60 000 training epochs. (c) Synaptic weights between the input to output neurons with 40 output neurons, when the NL ranges are $0 \sim 0.24$ and $0 \sim 0.77$. Reproduced from [51].

intervention, the side effects by abrupt depression can be mitigated slightly, but in the case of unsupervised learning without external control, if abrupt depression of the synaptic weight occurs, the learned weights may be lost momentarily. In the case of supervised learning, however, the possibility of weight loss due to abrupt depression can be reduced by external intervention. It is necessary to make an effort to improve the device structure and conductance change mechanism in order to prevent abrupt depression. Also, the endurance characteristic of the memory devices is an important factor in the learning process, and the retention characteristic is an important factor in the inference process. This is because the inference accuracy is affected by how long the synaptic device can maintain the weight state determined through the learning process. Considering actual chip implementation, it will be important to analyze and improve endurance and retention characteristics in synaptic devices. In addition, the analysis of the uniformity of the device is critical for large-scale chip implementation. In general, neural networks are known to have some degree of immunity to device variation [30, 52]. However, the variation between synaptic devices should be as small as possible, because it negatively affects power consumption and speed in the learning process. Recently, studies on the implementation of an HNN using a proven flash memory technology have been actively conducted due to the immaturity of the new memory technologies [37, 53, 54].

2.2. Software-based learning algorithm (BP)

The software-based deep neural networks (SW-DNNs) with a well-studied BP algorithm [55] have shown excellent performance. As shown in figure 5, a SW-DNN consists of the input layer, hidden layers and output layer. Each node of the layer is known as a neuron and a node connection between adjacent layers is called a synapse. The strength of a synapse is the synaptic weight, or simply the weight. By using the BP algorithm, we can find the optimal values of the weights to

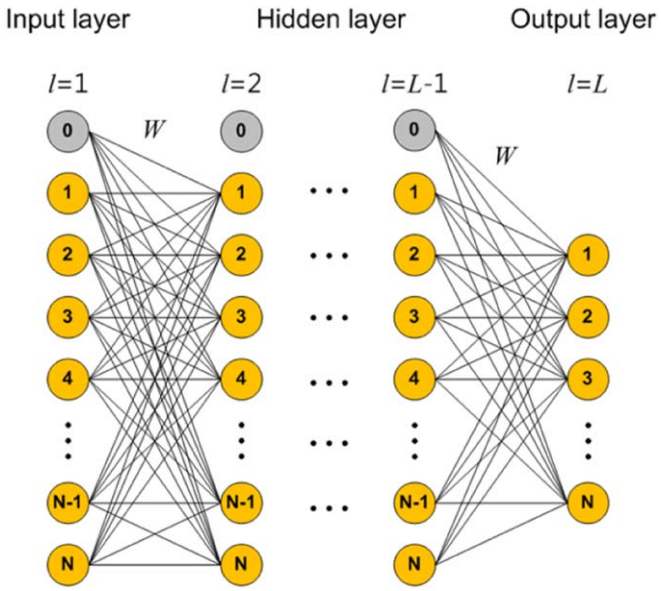

Figure 5. Typical structure of the DNNs composed of the input layer, hidden layers and output layer.

minimize the training and generalization errors. In the process of finding the optimal weights, the vector-by-matrix multiplication (VMM) of forward propagation (FP) and BP accounts for a large portion of the computational tasks. However, HW-DNNs can perform this VMM with very low power and high speed because the result of the VMM is simply the current of the electronic synaptic device array, which is the product of the input voltage and conductance. There are two main approaches for HW-DNNs: on-chip training and off-chip training. We define on-chip training where the weights are updated within the synaptic device array for each iteration, not depending on batch size. For on-chip training, the hardware, including the synaptic device array, should perform FP, BP and weight updates. On the other hand, off-chip training means the weight updates are performed by software and then the calculated weights are transferred to the synaptic device array. In this case, the synaptic array is only used for the VMM for FP after training, which is also called inference or dot-product engine (DPE). Table 3 shows the off-chip and on-chip training rule of an HW-DNN compared to an SW-DNN. In an HW-DNN, two identical electronic devices are required to represent a unit synapse, because the weights (W_{ij} for the weight of the synapse between the ith neuron in the l-1 layer and the jth neuron in the l layer) of the unit synapse in neural networks should have both positive and negative values. The input signal (a_i^{l-1}) (for the ith neuron in the l-1 layer) and the weight (W_{ij}) can be represented by the applied voltage (V_i^{l-1}) and conductance difference of the unit synapse ($G_{ij}^{+} - G_{ij}^{-}$), respectively. The positive and negative values of the weights can be expressed by subtracting the output current from a pair of synapses ($W_{ij} = G_{ij}^{+} - G_{ij}^{-}$). By connecting all unit synapses in the l-1 layer connected to the jth neuron in the l layer, the currents from each unit synapse are summed ($\sum_N^i (G_{ij}^{+} - G_{ij}^{-}) V_i^{(l-1)}$). This weighted-sum value (s_j^l) is then converted to the input signal of the next layer (a_i^l) using an activation function (f), which is implemented by electronic circuits. For off-chip training, the synaptic device array is responsible for this FP. Furthermore, for on-chip training, BP and weight updates should be implemented using the synaptic device array. In addition to the FP, to compute the BP, the backward-weighted-sum should be performed after the weight matrices are transposed. In other words, the postsynaptic neurons during FP should act as the function of the presynaptic neurons in the BP, and vice versa. That is, the synaptic device array should be transposable to implement BP. The input signal in the backward direction (δ_j^l for the ith neuron in the l layer) and the weight (W_{ij}) are represented by applied voltage (V_j^l) and the conductance difference in the unit synapse ($G_{ij}^{+} - G_{ij}^{-}$), respectively. In this way, the backward-weighted sum ($\sum_j^M W_{ij} \delta_j^l$) can be performed by connecting all unit synapses in the l layer connected to the ith neuron in the l-1 layer. Then, we can obtain the error delta value of the ith neuron in the l-1 layer (δ_j^{l-1}) by multiplying the derivative value of the activation function $f'(s_i^{l-1})$ by the backward-weighted-sum value. After the error delta values of all layers excluding the input layer are obtained through this process, the weights can be updated according to the product of the learning rate (η), error delta value of the postsynaptic neuron (δ_j^l) and activated value of the presynaptic neuron ($f(s_i^{l-1})$).

2.2.1. Off-chip training. A nanoscale memory crossbar array can naturally carry out VMM, which is a computationally expensive task for many important applications in a single time step by Kirchhoff's current law [56]. Some researchers use on-chip training schemes to minimize the effect of device variation on learning accuracy [57–60]. However, up-to-date on-chip training approaches are slow, because of iterative processes with extensive reading and writing of all devices, with limited performance/energy efficiency compared to software [61] and potential device wearout [62]. To overcome these issues, some groups developed off-chip training in neuromorphic computation as a realistic solution.

1) UCSB's three-layer perceptron network using NOR flash memory.

   Merrikh-Bayat *et al* implemented a prototype three-layer neuromorphic network using arrays of highly optimized embedded nonvolatile floating-gate cells redesigned from a commercial 180 nm NOR flash memory (figures 6(a)–(c)) [63]. Their main advantage is very mature fabrication technology. Custom design has been recently demonstrated for the industrial-grade 180 [64, 65] and 55 nm [66] NOR flash memories. Floating-gate cells are quite suitable to be used as adjustable synapses in neuromorphic computing, as the memory arrays are redesigned to allow for individual, precise adjustment of the memory state of each device. In this network design, the energy-saving gate coupling [67, 68] of the peripheral and array cells, which works well in the subthreshold mode was used, with a nearly exponential dependence of the drain current I_{DS} of the memory cell on the gate voltage V_{GS} (figure 6(d)). The desirable synaptic weights calculated in an external computer running a similar 'precursor' software-implemented network, using the standard BP algorithm,

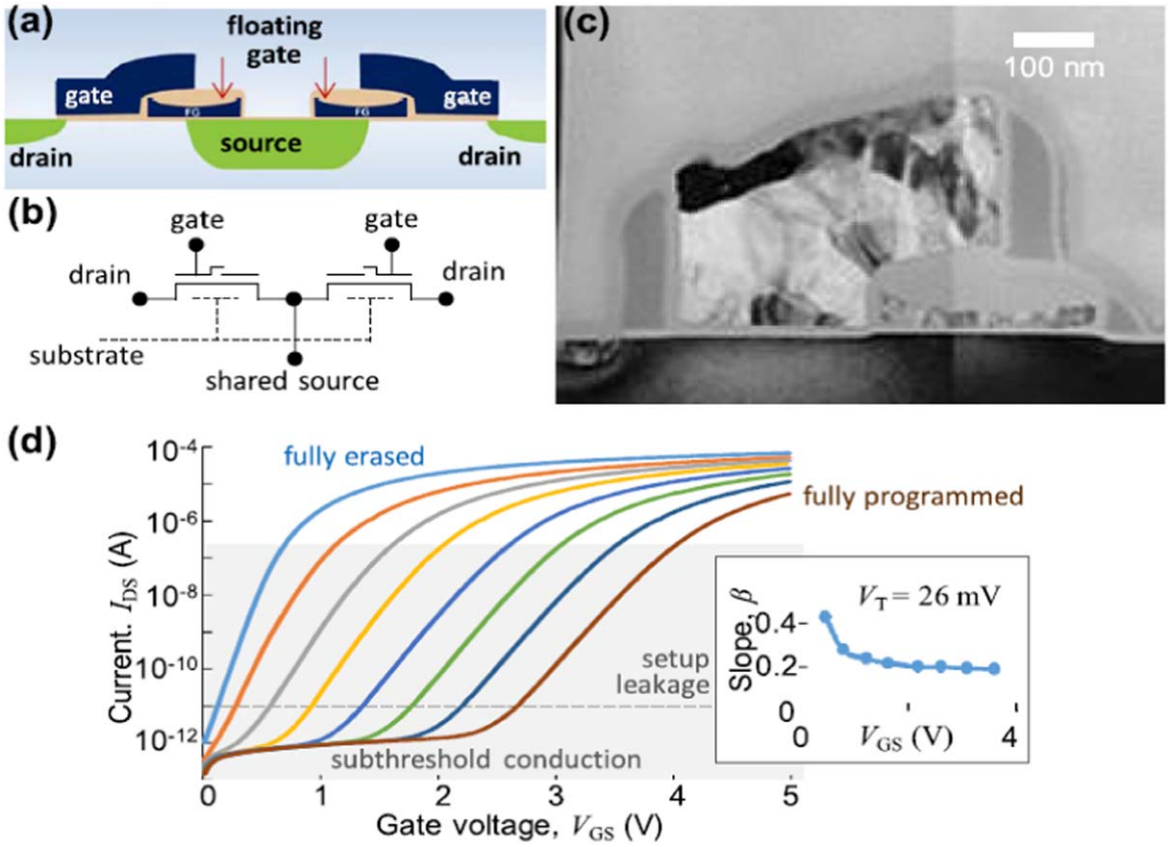

Figure 6. ESF1 NOR flash memory cells. (a) Cross-section of the two-cell 'supercell' (schematically) and (b) its equivalent circuit. (c) TEM cross-sectional image of one memory cell, fabricated in a 180 nm process. (d) Drain current of the cell as a function of the gate voltage, at $V_{DS} = 1$ V, for several memory states. (d) Gray-shaded region shows the subthreshold conduction region; currents below $I_{DS} = 10$ pA (the level shown with the dashed line) are significantly contributed by leakages in the experimental setup used for the measurements. Inset: extracted slope of this semilog plot, measured at $I_{DS} = 10$ nA, as a function of the memory state (characterized by the corresponding gate voltage). © [2017] IEEE. Reprinted, with permission, from [63].

Table 3. Off-chip and on-chip training rule of HW-based neural networks.

		Hardware based	
Target	Software based	Off-chip	On-chip
Weights $\boldsymbol{W_{ij}}$	$\boldsymbol{W_{ij}}$	$\boldsymbol{G_{ij}^{+} - G_{ij}^{-}}$	$\boldsymbol{G_{ij}^{+} - G_{ij}^{-}}$
FP $\boldsymbol{S_j^{(l)}}$	$\sum_i^N \boldsymbol{W_{ij} a_i^{(l-1)}}$	$\sum_i^N (\boldsymbol{G_{ij}^{+} - G_{ij}^{-}}) \boldsymbol{V_i^{(l-1)}}$	$\sum_i^N (\boldsymbol{G_{ij}^{+} - G_{ij}^{-}}) \boldsymbol{V_i^{(l-1)}}$
Activated value $\boldsymbol{a_j^{(l)}}$	$\boldsymbol{f}(s_j^{(l)})$	$\boldsymbol{f}(s_j^{(l)})$	$\boldsymbol{f}(s_j^{(l)})$
BP $\boldsymbol{\delta_i^{(l-1)}}$	$\sum_j^M \boldsymbol{W_{ij} \delta_j^{(l)}} \cdot \boldsymbol{f}'(s_i^{(l-1)})$		$\sum_j^M (\boldsymbol{G_{ij}^{+} - G_{ij}^{-}}) \boldsymbol{V_j^{(l)}} \cdot \boldsymbol{f}'(s_i^{(l-1)})$
Weight updates $\boldsymbol{\Delta W_{ij}}$	$-\boldsymbol{\eta} \cdot \boldsymbol{\delta_j^{(l)}} \cdot \boldsymbol{f}(s_i^{(l-1)})$		$-\boldsymbol{\eta} \cdot \boldsymbol{\delta_j^{(l)}} \cdot \boldsymbol{f}(s_i^{(l-1)})$

were transferred into the network by analog tuning of the memory state of each floating-gate cell, with peripheral analog demultiplexer circuitry. Only one cell of each pair, corresponding to a particular sign of the weight value, was tuned, while its counterpart was kept at a very small, virtually zero, initial conductance to decrease the weight transfer time. The digital encoders and shift register circuits and their layouts were synthesized from Verilog in a standard 1.8 V digital CMOS process. All active blocks of the circuit, including 101 780 floating-gate cells, have a total area below 1 mm^2. The network has shown a 94.7% classification fidelity on the MNIST dataset benchmark, close to the 96.2% obtained in simulation. The classification of one pattern takes a sub-1 μs time and sub-20 nJ energy—both numbers much better than those in the best reported digital implementations of the same task.

2) Arizona State University's BNN based on RRAM.

Using an RRAM-based synapse, Yu *et al* [69] demonstrated BNN with BP with a new record for the scale of the synaptic array up to 512 × 1024. They

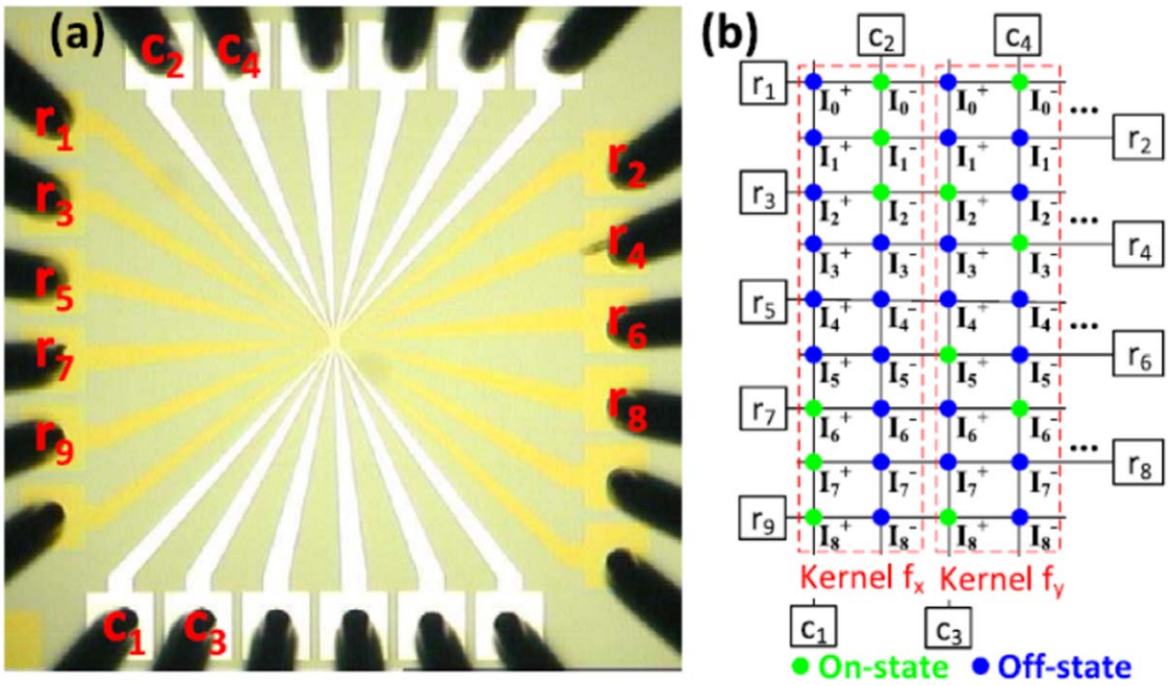

Figure 7. (a) Microscopic top-view image of a fabricated 12 × 12 cross-point array. Probe card tips have touched on the pads. (b) Implementation of the Prewitt horizontal kernel (f_x) and vertical kernel (f_y). © [2016] IEEE. Reprinted, with permission, from [71].

proposed a methodology to binarize the neural network parameters with the goal of reducing the precision of weights and neurons to 1-bit for classification. They experimentally demonstrate the BNN on Tsinghua's 16 Mb RRAM macrochip fabricated in a 130 nm CMOS process. Because of the negative value of weights in the BNNs, they used two columns to represent the weight by taking the differential output. Even under finite bit yield and endurance cycles, the system performance on the MNIST handwritten digit data set achieves ∼96.5% accuracy for classification, close to ∼97% accuracy by the ideal software implementation. In addition, they showed redundant and massively parallel networks provided high resilience to random bit errors. This work reported the largest scale of the synaptic arrays and achieved the highest accuracy so far. The proposed BNN implementation is also applicable to the neuromorphic designs with other binary memories such as static RAM (SRAM), PCM and even spin-transfer torque magnetic randon access memory (STT-MRAM).

3) HP's DPE based on a memristor crossbar.

Hu *et al* [70] demonstrated high-precision analog tuning and control of memristor cells across a 128 × 64 array and evaluated the resulting VMM computing precision. Utilizing the natural current accumulation feature of the memristor crossbar, the DPE was developed as a high-density, high-power efficiency accelerator for VMM. Single-layer neural network inference is performed in 128 × 64 arrays, and the performance compared to a digital approach is assessed. They invented a conversion algorithm to map arbitrary matrix values appropriately to memristor conductance in a realistic crossbar array, accounting for device physics and circuit issues to reduce computational errors. This conversion algorithm can be extended to any other crossbar structures or cross-point devices by just replacing the circuit or device models. Accurate device resistance programming in large arrays is demonstrated by close-loop pulse tuning and access transistors. To validate this approach, they simulated and benchmarked one of the state-of-the-art neural networks for pattern recognition on the DPEs. The result shows no accuracy degradation compared to the software approach (99% pattern recognition accuracy for the MNIST data set) with only 4-bit DAC/ADC requirement, while the DPE can achieve a speed-efficiency product of 1000 × to 10 000 × compared to a custom digital application-specific integrated circuit.

4) Arizona State University's convolution kernel operation on resistive cross-point array.

Gao *et al* [71] proposed a dimensional reduction of a 2D kernel matrix into a 1D column vector, i.e. a column of the array, and enabled the parallel read-out of multiple 2D kernels simultaneously. They experimentally demonstrated the convolution kernel in the CNN on a 12 × 12 HfO_x crossbar. Figure 7 shows the microscopic top-view image of our fabricated 12 × 12 cross-point array: six contact pads are located at the four edges, and resistive Pt/HfO_x/TiN stacks (from top to bottom) are formed at the cross-point junctions of the rows and columns. As a proof-of-concept demonstration, they used the Prewitt kernels to detect both the horizontal and vertical edges of the 20 × 20 pixels of the black and white MNIST handwritten digits data set. The experiments were performed on the fabricated 12 × 12 resistive cross-point array based on the Pt/HfO_x/TiN structure. The experimental

results of the Prewitt kernel operation perfectly match the simulation results, indicating the feasibility of the proposed implementation methodology of the convolution kernel on resistive cross-point array. The proposed methodology can be applied to implementing larger kernels in deeper CNN architecture on-chip.

2.2.2. On-chip training. In contrast to off-chip training, hardware including the synaptic device array should perform BP and weight updates as well as FP for on-chip training. However, there are some additional considerations when implementing on-chip training with hardware. Since the VMM by synapse array is an analog computation, it is important to determine how to implement the peripheral circuits that control the synapse array (analog or digital). While analog peripheral circuits can power-efficiently interact with the synapse array, digital peripheral circuits can provide high precision. In addition, when updating the weights, it is important to update considering the non-ideal characteristics of the synaptic device. Most electronic synaptic devices have nonlinear and asymmetric conductance response, and there are many reports that these non-ideal characteristics can degrade the training accuracy [57, 72, 73]. In addition, the number of conductance levels that can be represented by synaptic device is limited, and thus exhibits low precision compared to the weights that can be expressed in software. Even the analog peripheral circuits that control the electronic synaptic device arrays have non-ideal effects, which result in degradation of the neural networks [74]. To enable low-power operation while handling the non-ideal effects of devices, various approaches using modified weight-update rules have been attempted, such as crossbar-compatible weight update and Manhattan weight update [57, 59]. It also includes simplification of the activation functions that can be implemented with simple electronic circuits [75]. Due to these non-ideal effects and simplification of the BP rule, the performance of HW-DNNs for on-chip training still remains in simple learning tasks such as the MNIST data set. Nonetheless, the HW-DNN for on-chip training is important because it has tolerance to the device variation and can be constantly trained. There have been several studies of the implementation of HW-DNNs for on-chip training, and we review these.

1) IBM's 500 × 661 PCM array.

Burr *et al* used 2-PCM devices as a synapse and implemented a three-layer perceptron network [57]. The neural network consists of 528 input neurons, 250 neurons in the first hidden layer, 125 neurons in the second hidden layer, and ten output neurons, and was trained by a subset (5000 examples) of the MNIST data set. The size of the fabricated PCM array is 500 × 661, and the total number of PCM devices used for neural networks including bias neurons is $(529 \times 250 + 251 \times 125 + 126 \times 10) \times 2 = 329\,770$. The weights represented by the conductance of the PCM devices are updated by the BP algorithm and crossbar-compatible weight-update method. However, the VMM is performed by software. Since the weight update is determined by the product of the error delta value and presynaptic neuron value, how to perform this computation efficiently is important. In a crossbar array, when a presynaptic neuron value and an error delta value are inputted to one node and another node, respectively, the conductance can be immediately changed by the voltage difference between the two nodes. As shown in figure 8(a), they divided the magnitude of presynaptic neuron and error delta values by four pulses, so the conductance is changed by the time overlap between two pulses, representing presynaptic neuron and error delta values, respectively. When the crossbar-compatible weight-update method is used, the classification accuracy does not degrade compared with the case of using software-based conventional weight update (figure 8(b)). However, it has been shown that the nonlinearity and asymmetry of the conductance response of the electronic synaptic device can significantly degrade the classification accuracy (figure 8(c)). To mitigate such asymmetry of the conductance response, they proposed the occasional reset process with enough frequency. In addition, they simulated the effects of various non-idealities of the conductance response by simulation, such as stochasticity, variable maxima and the presence of nonresponsive (conductance is not changed by the applied pulses) devices. In their later works, additional research was conducted on the training power and speed estimation, design considerations of peripheral circuits and other factors that could affect the implementation of the HW-DNNs. [57, 75–78].

2) UCSB's 12 × 12 memristive crossbar array.

Prezioso *et al* fabricated a memristive crossbar array with 12 × 12 devices, and implemented a perceptron network [59]. At each cross point, an Al_2O_3/TiO_{2-x} memristor was used with low variability. The perceptron network consists of ten input neurons and three output neurons including a bias neuron in the input layer, and is trained by 30 binary images consisting of 3 × 3 pixels. As in the previously reviewed work, two devices are required for a synapse, so the total number of synapses is $10 \times 3 \times 2 = 60$. This synapse array performed the VMM, and the weight updates are also conducted within the synapse array. Since the only hardware is the synapse array, the synaptic current was measured for each column of the synapse array, and then the subtraction of the synaptic current between adjacent columns was performed by external electronics (figure 9(a)). For weight update, they applied a Manhattan update rule [79]. In this rule, whether to increase or decrease the weights is determined by the sign of the product of the presynaptic neuron and error delta values (figures 9(b) and (c)). It is easy to implement in hardware because only a single pulse is required for each weight update. Due to the non-ideal characteristics of the synaptic device, such as nonlinearity of the conductance response, applying multiple pulses for accurate weight update can be impractical in actual electronic devices [80]. Although the size of the neural networks is small and the training

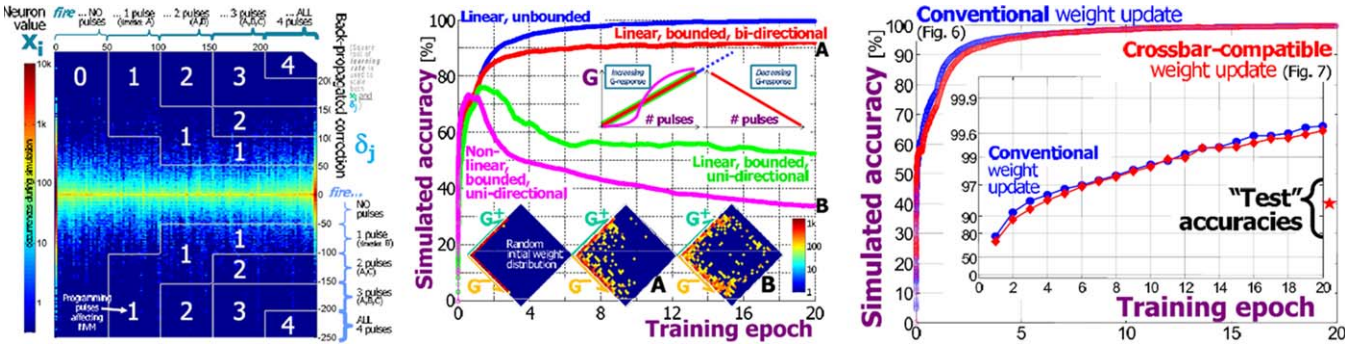

Figure 8. (a) In a crossbar, efficient learning requires neurons to update weights in parallel, firing pulses whose overlap at the various NVM devices implements training. (b) Computer neural network simulations show that a crossbar-compatible weight-update rule is just as effective as the conventional update rule. (c) Bounding G values reduces neural network training accuracy slightly, but uni-directionality and nonlinearity in G-response strongly degrade accuracy. Figure insets map NVM-pair synapse states on a diamond-shaped plot of G^+ versus G^- (weight is vertical position) for a sampled subset of the weights. © [2014] IEEE. Reprinted, with permission, from [57].

set is simple, they have shown well-trained results using a fabricated memristive crossbar array.

3) Tsinghua's 128 × 8 RRAM array.

Yao *et al* demonstrated a perceptron network for face recognition with a 128 × 8 RRAM array [81]. The array consists of a RRAM with a $TiN/TaO_x/HfAl_yO_x/TiN$ stack and transistor (1T-1R). This RRAM stack showed a bi-directional conductance response with respect to the number of applied pulses, but it is nonlinear. The perceptron network was trained by nine grayscale face images, which are a cropped and subsampled subset of the Yale Face Database (figure 10) [82]. The training process is divided into two parts: inference and weight update. After the synaptic current is measured in the synapse array, it is applied to the software-based activation function. In the weight update, there are two weight-update methods: write-verify and without write-verify (figure 10). When using the write-verify method, the weight update is conducted considering the magnitude of the product of the presynaptic neuron and error delta values. Then, the calculated weight is stored as the conductance of the synaptic device through the verify process. On the other hand, when the method without the verify process is used, only the sign of the product of the presynaptic neuron and error delta values is considered, which is the Manhattan update rule. While the write-verify scheme could achieve higher classification performance, the without verify scheme could simplify the control system. As a result of face image classification, the accuracy rates were 88.08% and 85.04% for the write-verify scheme and the without verify scheme, respectively. In addition, the total latency of the without verify scheme is lower than that of the write-verify scheme, but the opposite result was shown in the case of energy consumption. This means that the write-verify scheme needs more programming pulses at each epoch, but fewer iterations are required for training.

2.2.3. Requirements.

1) Nonlinearity and asymmetry.

According to the reported papers, nonlinearity and asymmetry of the conductance response are important factors when performing on-chip training [57, 83–89]. However, there is no standard and clear measure to compare nonlinearity in various synaptic devices in which the conductance varies nonlinearly with the applied pulses. Here, we improve the conventional model in [30] to better represent the conductance response of synaptic devices. To obtain a model that fits the conductance response of synaptic devices, we use the equations that express the conductance change (δG) in [30]. They are expressed as follows for long-term potentiation (LTP) and long-term depression (LTD), respectively:

$$\delta G_{\mathrm{p}} = \alpha \exp\left(-\beta \frac{G - G_{\min}}{G_{\max} - G_{\min}}\right) \text{in LTP}, \tag{1}$$

$$\delta G_{\mathrm{d}} = -\alpha \exp\left(-\beta \frac{G_{\max} - G}{G_{\max} - G_{\min}}\right) \text{in LTD}, \tag{2}$$

where $G_{\max}$ and $G_{\min}$ are the maximum and minimum conductance values in the conductance response curves measured from a synaptic device, respectively. α is a fitting parameter and β is a nonlinearity factor. Because equation (1) represents the conductance change when pulse is applied in LTP, we can derive equation (3) as follows:

$$\delta Gp = \frac{G(n+1) - G(n)}{1} = \frac{\Delta G}{\Delta n} = \alpha \exp\left(-\beta \frac{G(n) - G_{\min}}{G_{\max} - G_{\min}}\right), \tag{3}$$

where n is the discrete pulse number. Here, we approximate equation (3) as follows to be transformed into the derivative form

$$\frac{dG}{dx} = \alpha \exp\left(-\beta \frac{G(x) - G_{\min}}{G_{\max} - G_{\min}}\right), \tag{4}$$

where x is the continuous pulse number. Although the pulse number (n) is basically a discrete value, we regard it as a continuous pulse number (x). Solving differential

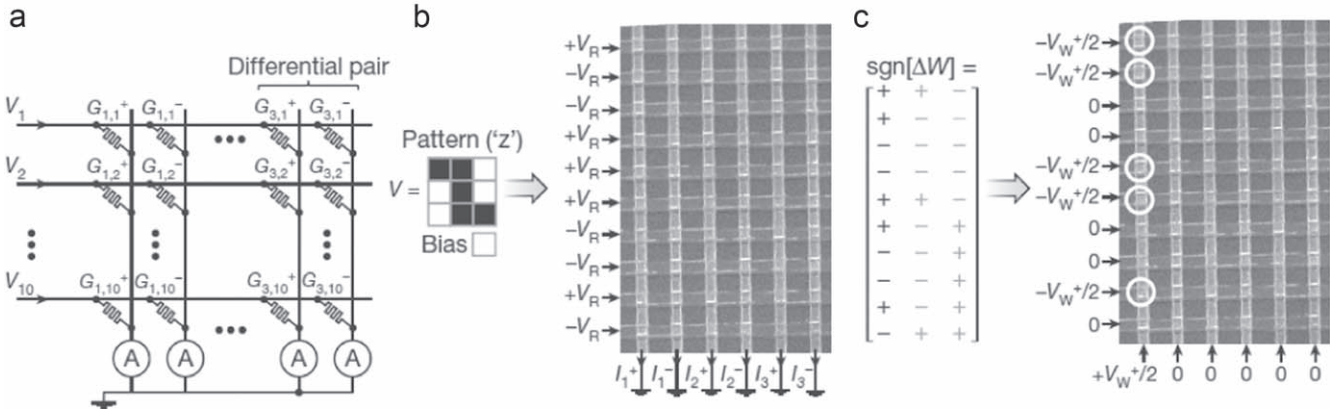

Figure 9. (a) Implementation of a single-layer perceptron using a 10 × 6 fragment of the memristive crossbar. (b) Example of the classification operation for a specific input pattern (stylized letter 'z'), with the crossbar input signals equal to $+V_R$ or $-V_R$, depending on the pixel color. (Read and write biases were always $V_R = 0.1$ V and $V_W^{\pm} = 1.3$ V, respectively) (c) Example of the weight adjustment in a specific (first positive) column, for a specific error matrix. At the step shown, only the synapses whose weights should be increased (marked by '1' in the table on the left) are adjusted, that is, the memristor conductances $G_{1,1}^{+}$, $G_{1,2}^{+}$, $G_{1,5}^{+}$, $G_{1,6}^{+}$ and $G_{1,9}^{+}$ are being increased. [59] (2015) © 2018 Springer Nature Limited. All rights reserved. With permission of Springer.

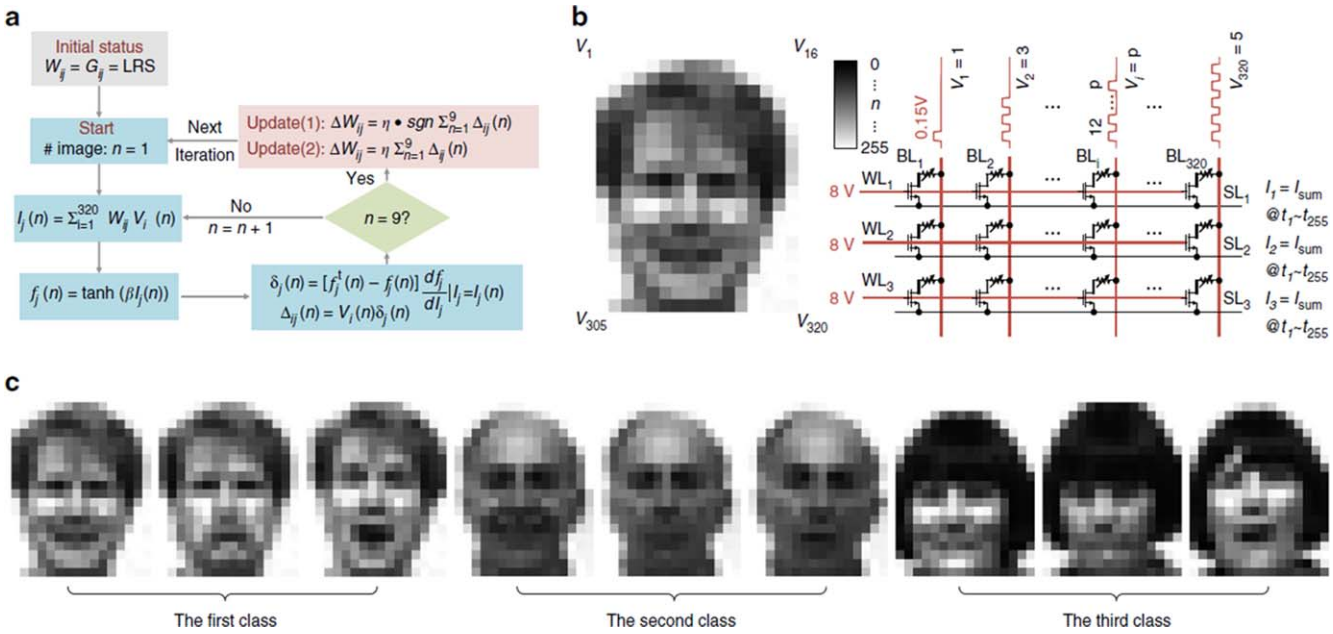

Figure 10. (a) Training process flow chart. In this demonstration, a batch learning model is used to accelerate the converging speed. Here, '*n*' represents the number of the pattern, ranging from 1–9, '*i*' implies the index of a pixel of an input pattern and can be defined from 1–320, '*j*' is the number of output neurons, that is 1–3. Correct classification during the inference phase means the active function value of a matching class of the input pattern is greater than the other two classes. This network converges when all training patterns are correctly recognized. (b) Schematic of parallel read operation and how a pattern is mapped to the input. (c) Nine training images, which are a cropped and subsampled subset of the Yale Face Database. Reproduced from [81]. CC BY 4.0.

equation (4), the conductance response in LTP can be expressed as a function of the continuous pulse number (x) as follows:

$$G_{LTP}(x) = G_{\min} + \frac{G_{\max} - G_{\min}}{\beta} \ln\left(\frac{\alpha\beta}{G_{\max} - G_{\min}}\right) + \frac{G_{\max} - G_{\min}}{\beta} \ln\left(x - 1 + \frac{G_{\max} - G_{\min}}{\alpha\beta}\right) = a + \frac{1}{\beta}\ln(x + c). \quad (5)$$

In the same way, we can obtain the equation for LTD as follows:

$$G_{LTD}(x) = G_{\max} - \frac{G_{\max} - G_{\min}}{\beta} \ln\left(\frac{\alpha\beta}{G_{\max} - G_{\min}}\right) - \frac{G_{\max} - G_{\min}}{\beta} \ln\left(x - 1 + \frac{G_{\max} - G_{\min}}{\alpha\beta}\right). \quad (6)$$

Equations (5) and (6) show an improved model from the model in [30] and explain the LTP and LTD conductance

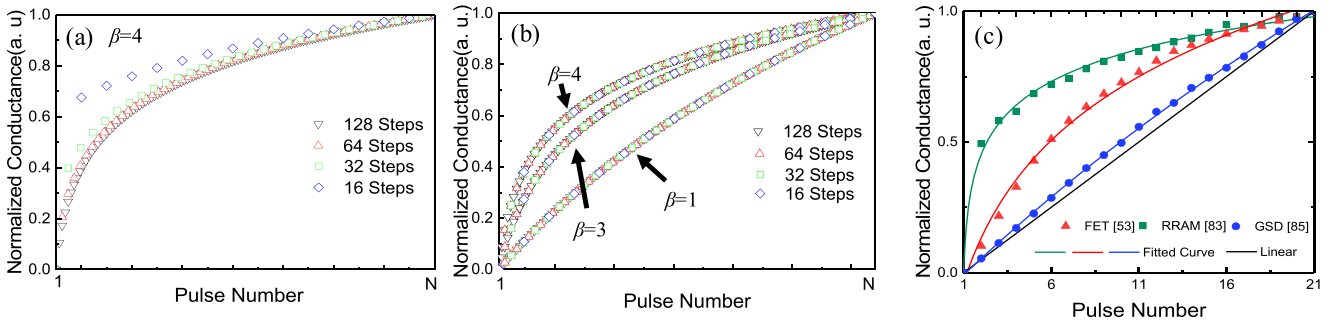

Figure 11. (a) Conductance responses with respect to the number of conductance levels when β is 4, in the case of using conventional models in [30]. (b) Conductance responses with respect to the number of conductance levels when β is 1, 3 and 4, in the case of using the improved model. (c) When using the improved model, conductance response for several types of synaptic devices, especially RRAM [83], FET-type [53], gated Schottky diode (GSD) [85] and ideal linear conductance response.

response, respectively. Since the conventional model in [30], expressed as equations (1) and (2), represents only the conductance change at a certain conductance state, it is not suitable for representing whole conductance response. When the conductance response is represented by using the conventional model, the nonlinearities are different according to the number of conductance levels even if β is the same (figure 11(a)). However, when equation (5) is used for representing the conductance response, the nonlinearities are the same if β is the same regardless of the number of conductance levels (figure 11(b)). Equation (5) also successfully fits the conductance response of various device types, as shown in figure 11(c) [83].

When the conductance response is nonlinear, it is difficult to perform accurate weight updates and this leads to loss of accuracy. Thus, there has been a report that conductance response should be linear for on-chip training performance, as shown in figures 12(a) and (b). Subsequently, a device with near-linear conductance response when the identical pulses are applied was reported, as shown in figures 12(c) and (d) [85]. Another paper has reported that the conductance response can be linearly improved by adding a resistor to the synaptic device, as shown in figure 12(e) [88]. Similarly, asymmetry of the conductance response between LTP and LTD is also an important parameter because it can degrade the on-chip training performance [57]. However, most HW-DNNs use two devices per synapse to represent a negative to positive weight, so only the LTP or LTD curve of the devices can be used to improve the asymmetry.

In contrast to on-chip training, nonlinearity is not such an important issue when performing off-chip training. It has been reported that the precise adjustment of weights can be individually programmed in the case of NOR flash [90] with mixed signal processing and tunable conductance. In addition, even for RRAMs with nonlinear conductance response, precise tuning can be used to adjust the weight accurately (less than 1%) without considering nonlinearity [91]. The high classification accuracy of the MNIST data set was obtained with this method. However, if the neural networks are very large with very deep layers, the method might be a challenge to adjust the weights to synaptic devices individually. As an alternative to the precise tuning of entire weights with high precision, the 1-bit weights can be used without significant accuracy loss for the MNIST dataset classification [69].

2) Weight precision.

The necessary weight precisions when performing on-chip training are somewhat different for each report [69, 92, 93], as shown in figure 13. However, a weight precision of more than 5-bit (32 levels) is commonly required to implement on-chip training without significant degradation of accuracy. By having stable and high weight precision from the minimum to maximum conductance makes it possible to update the weight more finely, which is related to the learning rate of the BP algorithm.

In off-chip training, there is a method to precisely tune the conductance of NOR flash or memristive devices [90, 91]. However, peripheral circuits are required for tuning precise conductance to synaptic devices, especially for sensing the current. The peripheral circuit can consume a lot of power and time in very large DNNs. On the other hand, BNN which has 1-bit weight precision was reported, as shown in figure 14 [69]. The weights are trained first with 32-bit floating point precision in software, and then the weights are binarized with small accuracy loss in the case of the MNIST dataset classification. However, the performance of BNNs can still be a challenge in large-scale neural networks [63].

3) Dynamic range.

Dynamic range is defined as the ratio between the maximum and minimum conductance of the synaptic devices. It is not necessary to use the on/off ratio of synaptic devices as the dynamic range. In other words, the dynamic range can be a specific range of conductance used to represent the weights. When implementing on-chip training, high weight precision is required for better performance of HW-DNNs. If the conductance response of a synaptic device has a large dynamic range, then the number of conductance levels (weight precision) can be large. Similarly, in off-chip training, the conductance of synaptic devices can also be finely tuned within the large dynamic range when

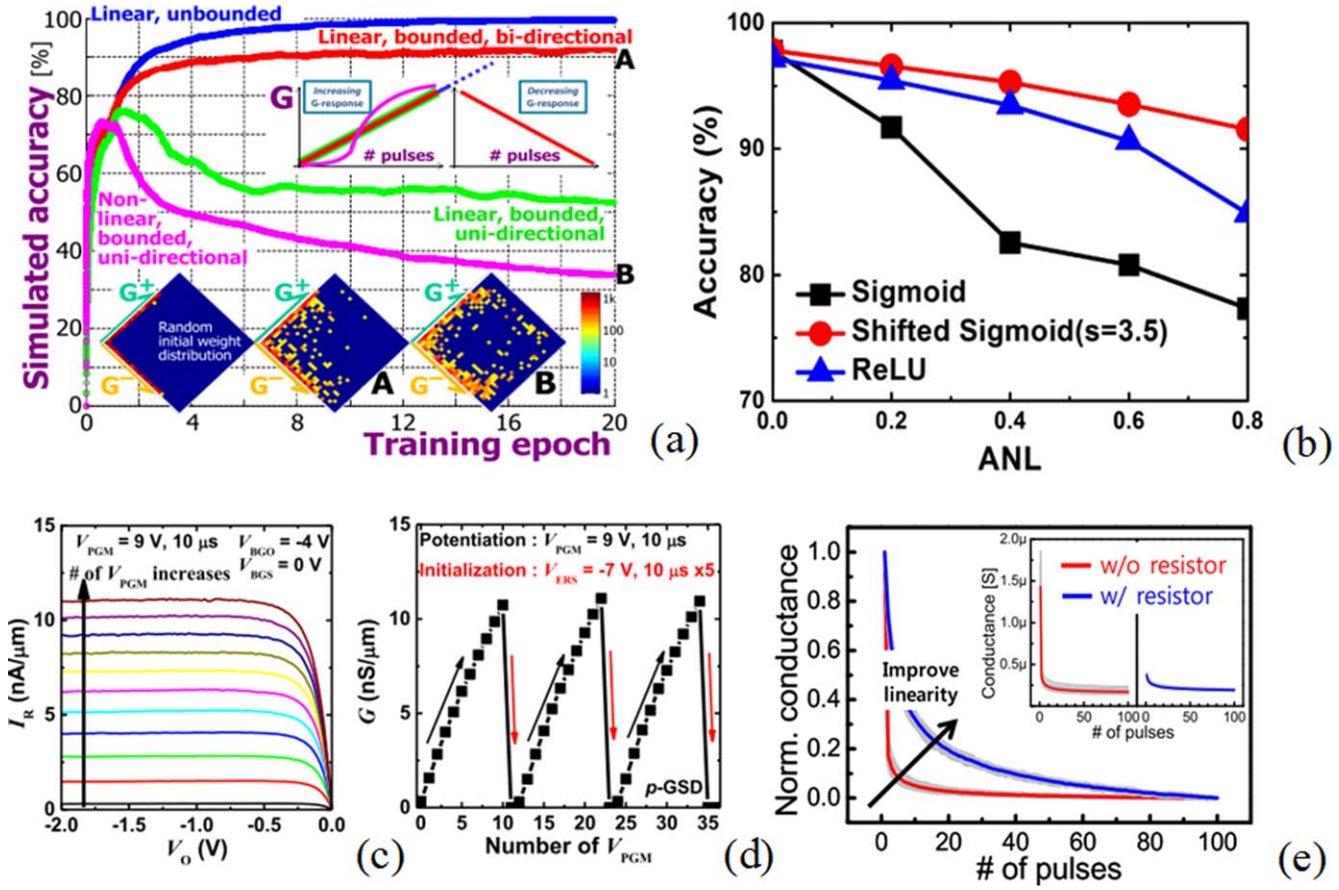

Figure 12. Researches of a neural network affected by nonlinearity of synaptic devices and to improve nonlinearity. (a) and (b) show that linear conductance response has a good performance. (a) © [2014] IEEE. Reprinted, with permission, from [57]. (b) © [2018] IEEE. Reprinted, with permission, from [87]. (c) and (d) are *I–V* characteristics and conductance response of the GSD, respectively.© [2017] IEEE. Reprinted, with permission, from [85]. (e) shows that nonlinearity can be improved with an additional resistor. © [2017] IEEE. Reprinted, with permission, from [88].

transferring weights. The dynamic range can also be modified to lower the variation of synaptic devices, or have a more linear conductance response. Note that the endurance characteristic of a synaptic device can be degraded if the dynamic range is too large. Thus, the proper dynamic range (generally, max/min ratio more than ten and less than 100) for the characteristics of a synaptic device is required when performing on-chip and off-chip training [92].

4) Reliability: endurance, retention and variation.

The synaptic devices should endure as many pulses as possible for high learning performance. According to the reported papers [50, 69], if synaptic devices can endure more than 10000 pulses, there is no significant accuracy loss for the MNIST data set in online BNN, as shown in figure 15. Similar to endurance, the retention characteristics of a synaptic device can significantly affect the accuracy of on-chip training. If the weights of synaptic devices are changed during training or inference, this causes significant accuracy loss. Variation of synaptic devices such as device-to-device variation, cycle-to-cycle variation and pulse-to-pulse variation also affects the accuracy of the HW-DNNs for on-chip training, as shown in figure 16 [50, 94].

In addition, the aforementioned characteristics are important when performing off-chip training. The weights are pre-trained in software and then transferred to the synaptic device array. Therefore, the synaptic devices have to retain the information of pre-trained weights for a long time while enduring many pulses [50]. In addition, the several variations prevent the conductance of the device from being tuned to the expected value, which leads to a significant accuracy loss, as shown in figure 17 [90, 92].

5) Overall characteristics required for synaptic devices.

As mentioned above, table 2 summarizes the desirable performance metrics for synaptic devices [50]. As a synaptic device, it would be appropriate to have as small a dimension as possible, and as low an energy consumption as possible. However, other metrics such as the number of conductance levels (weight precision), dynamic range, retention and endurance can vary considerably depending on the learning algorithms (bio-inspired and software-based learning algorithms) and the applications (unsupervised, supervised, off-chip and on-chip training) of the neural networks. For example, the

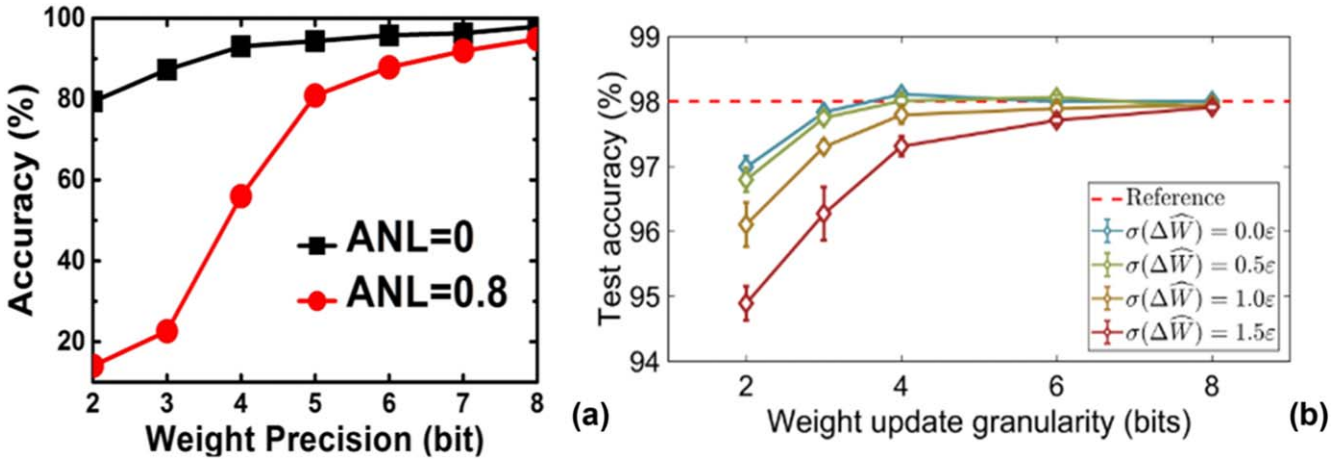

Figure 13. Accuracy with respect to the weight precision. © [2018] IEEE. Reprinted, with permission, from [87]. © [2018] IEEE. Reprinted, with permission, from [93].

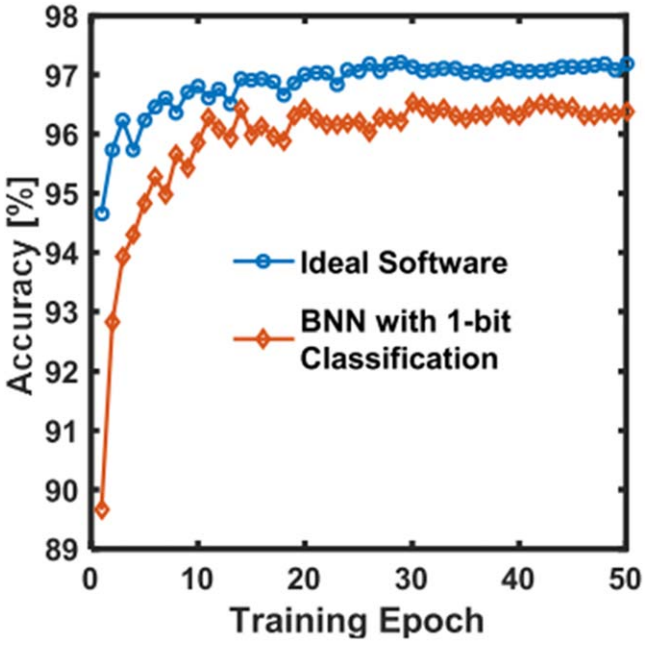

Figure 14. Accuracy of ideal software neural networks and binarized neural networks (BNNs). © [2016] IEEE. Reprinted, with permission, from [69].

linearity of the conductance response can be less important in off-chip training of HW-DNNs compared to on-chip training of HW-DNNs because iterative programming with a write-verify scheme can be used. In addition, these metrics cannot be independent of each other. If a pulse-to-pulse variation is high in a synaptic device, a required dynamic range needs to be larger than that in a synaptic device with a low pulse-to-pulse variation. In other words, a higher pulse-to-pulse variation requires a larger difference between adjacent conductance levels, resulting in a larger dynamic range in a synaptic device at the same number of conductance levels. A larger dynamic range more severely degrades the endurance characteristic, because a larger current, larger bias, or longer pulse width are required for the conductance change. Thus, the pulse-to-pulse variation needs to be as small as possible. Note that the number of program/erase cycling (endurance) of the conventional flash memory devices as an example is significantly limited because the threshold voltage shift is several volts (large). The endurance of a synaptic device can be significantly increased compared to that of conventional flash memory devices because relatively small conductance changes are required for each weight update. Therefore, it is important to design an HNN considering the learning algorithms, applications of the neural networks and inherent device characteristics according to the type of synaptic device.

2.3. DSNNs

SNN implementation has many advantages with regard to energy and latency compared to non-spiking implementation. However, because the training algorithms of SNN are less mature than those of SW-DNNs, SNN mimicking a biological algorithm such as STDP generally has low accuracy. To overcome this inferior accuracy of SNN, there have been some studies on the conversion from SW-DNNs to SNN and BP using spike [29, 95–100, 104].

2.3.1. Conversion from SW-DNNs to SNN. There are various methods of neural coding and neuron model that are used to convert SW-DNNs to SNN. In many studies, rate coding and temporal coding are usually used as neural coding because they are straightforward and simple information coding methods. In addition, I&F and LIF neuron models are usually used as a neuron model due to their biological plausibility. Several studies on this topic will be briefly introduced in the order of neuron models.

1) I&F neuron model. The I&F neuron model is the simplest neuron model in SNN. It sums input signal in membrane potential and fires when the membrane potential is higher than the threshold of a neuron. It is well known that the relation between the input spike rate and output spike rate

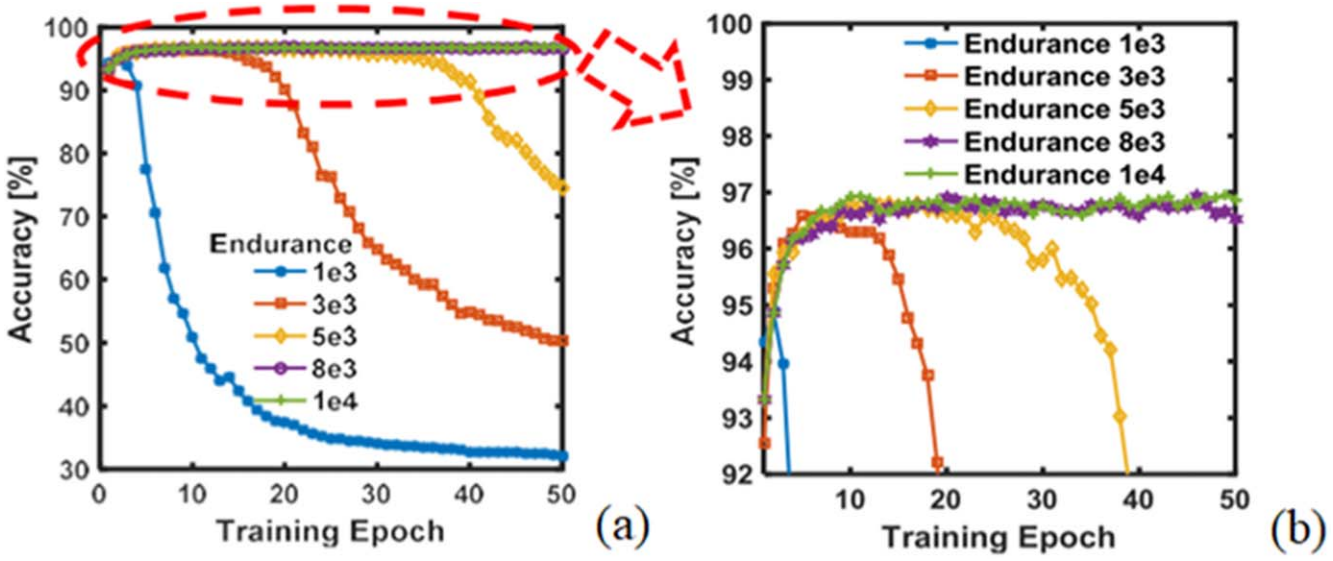

Figure 15. For on-chip training, (a) accuracy loss with different endurances of synaptic devices and (b) zoom-in of (a). © [2016] IEEE. Reprinted, with permission, from [69].

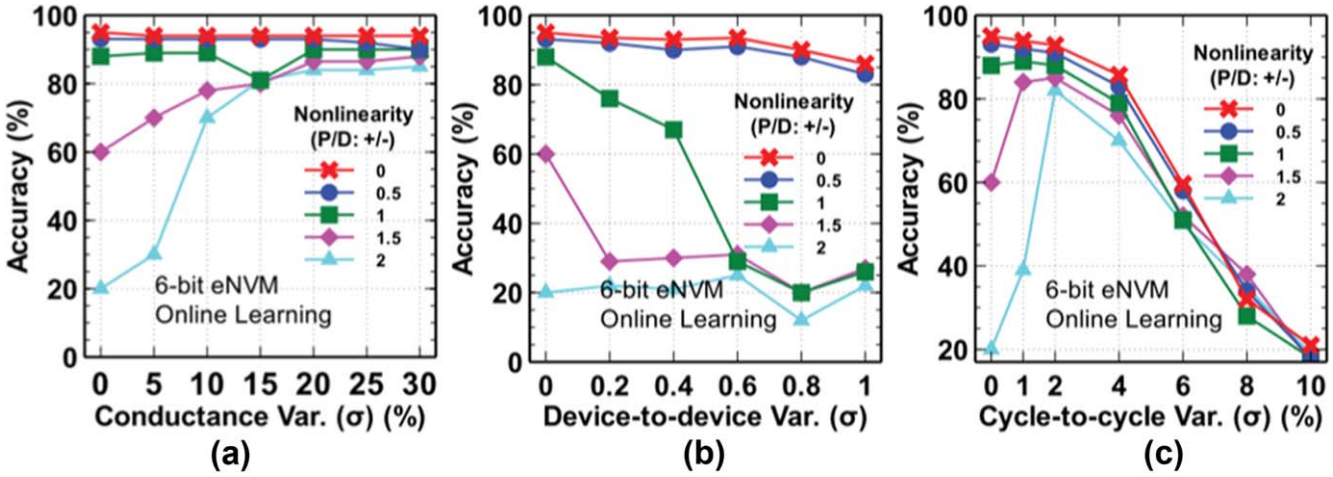

Figure 16. With 6-bit precision and different nonlinearity of synaptic devices, accuracy affected by (a) conductance variation, (b) device-to-device variation and (c) cycle-to-cycle variation. Linear devices are more tolerant than nonlinear devices. © [2018] IEEE. Reprinted, with permission, from [50].

in an I&F neuron using rate coding is a rectified linear unit (ReLU). Cao *et al* converted CNN to spiking CNN using I&F neuron and rate coding [95]. They used linear subsampling for the pooling layer and ReLU for an activation function. Conversion from SW-DNNs to SNN in this way results in accuracy drop because SW-DNNs and SNNs are not exactly the same in some aspects such as max firing rate and discrepancy of firing rate. Diehl *et al* proposed weight normalization to prevent the degradation of conversion accuracy [29]. The main idea of presented weight normalization is to rescale the weight so that the firing rate of each neuron is below the max firing rate. This can reduce the accuracy drop without searching the ideal hyperparameters in SNN. Rueckauer *et al* formulated the errors introduced by the reset method and reported implementation of the max-pooling layer, batch normalization and softmax using spiking neurons [96].

As rate coding requires several spikes to represent a single value, temporal coding is a more energy-efficient coding [97]. Because I&F neuron dynamics can only distinguish whether or not the input spikes precede the output spikes, it is not suitable for temporal coding. Therefore, Mostafa used I&F neuron with exponentially decaying synaptic current kernels [97]. He converted the exponential term generated by the exponentially decaying synaptic current kernel to the variable parameter. As a result, this gives a piecewise linear equation. Mostafa argued that temporal coding enabled fast processing because the network could be stopped after one of the output neurons created a spike.

2) LIF neuron model.

The LIF neuron model is similar to the I&F neuron model but has a leaky path at the membrane node. Hunsberger *et al* solved LIF neuron dynamics using rate

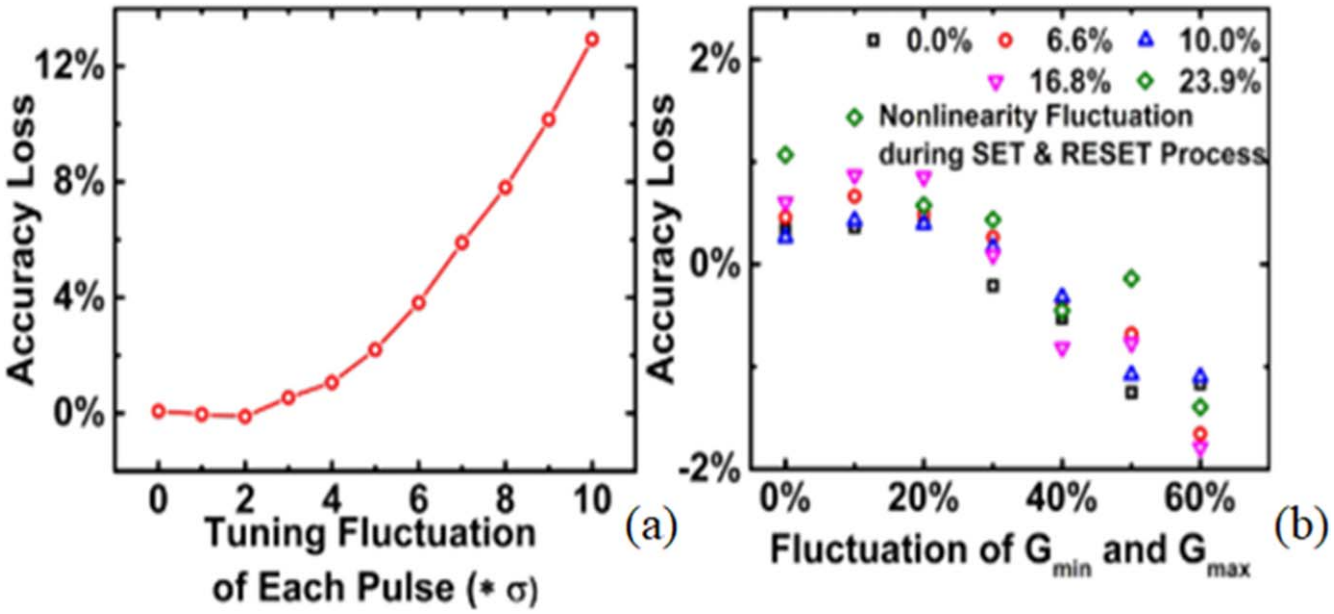

Figure 17. For off-chip training, accuracy loss with (a) pulse-to-pulse variation, (b) device-to-device variation of synaptic devices. © [2017] IEEE. Reprinted, with permission, from [92].

coding assuming constant input current and trained the SW-DNNs with noise [98]. The transfer function of SNN using rate coding and LIF neuron has an infinite derivative point. Because the infinite derivative point deteriorates the gradient descent training method, they suggested a soft LIF transfer function and used it for training. In addition, they showed that accuracy drop in conversion can be alleviated by using noise in training. Lee *et al* improved the accuracy by formulating the transfer function of LIF neurons from membrane potential and a series of spikes [99]. They formulated the transfer function of LIF neurons with WTA and showed 99.31% test accuracy in MNIST dataset test accuracy.

2.3.2. BP using spike. Neftci *et al* reported an event-driven random BP [100]. It uses LIF neuron with rate coding and translates the firing rate from the gradient descent formula to a spike-event-driven weight update. The weight update occurs whenever the input spike event happens by the number of errors propagated in the backward path. If the weights of the feedback path are the transpose of the weights of the forward path, such as the conventional BP, FP and BP cannot occur at the same time. Therefore, they used a feedback method called direct feedback alignment or skipped random BP [101, 102], which is a variant of random BP [103]. Error coding neuron generates spikes that are proportional to the difference between the output from the last layer and the actual correct answer, and the spikes are transmitted through a fixed random synapse directly connected to the neuron in each layer. In addition, Bengio *et al* presented equilibrium propagation, an energy-based neural network that uses Hopfield energy [104]. By explaining the equivalence of equilibrium propagation and STDP, they showed that a gradient descent can be performed in the SNN using the STDP algorithm.

3. Synaptic devices for implementing ANN

3.1. RRAM

Memristor is a two-terminal NVM device based on resistance switching [105]. Many kinds of memristors have been reported as a candidate for synaptic devices, since they are easy to construct into a matrix-type neural network. Among them, RRAM is a metal-insulator-metal (MIM)-type device having a switching characteristic through resistance change. That is, when different voltages are applied to each metal node, the current flows through the insulator and the amount of this current can be changed by the set and reset operations. Through the set operation, the resistance changes from high resistance to low resistance and changes from low resistance to high resistance through a reset operation. It is divided into unipolar and bipolar according to the polarity of the set and reset voltage (figure 18) [106]. Switching mechanisms vary widely depending on the type of insulating material, typically filamentary switching. The set process is attributed to the dielectric soft breakdown and creation of conductive filaments, usually consisting of oxygen vacancies. The reset process is attributed to the rupture of the conductive filament, usually by the recombination of oxygen vacancies with oxygen ions that migrated from the electrode/oxide interfacial reservoir [107].

From a neuromorphic system point of view, RRAM is attractive in terms of simple structure, low power, CMOS compatibility and scalability for high density. Seo *et al*

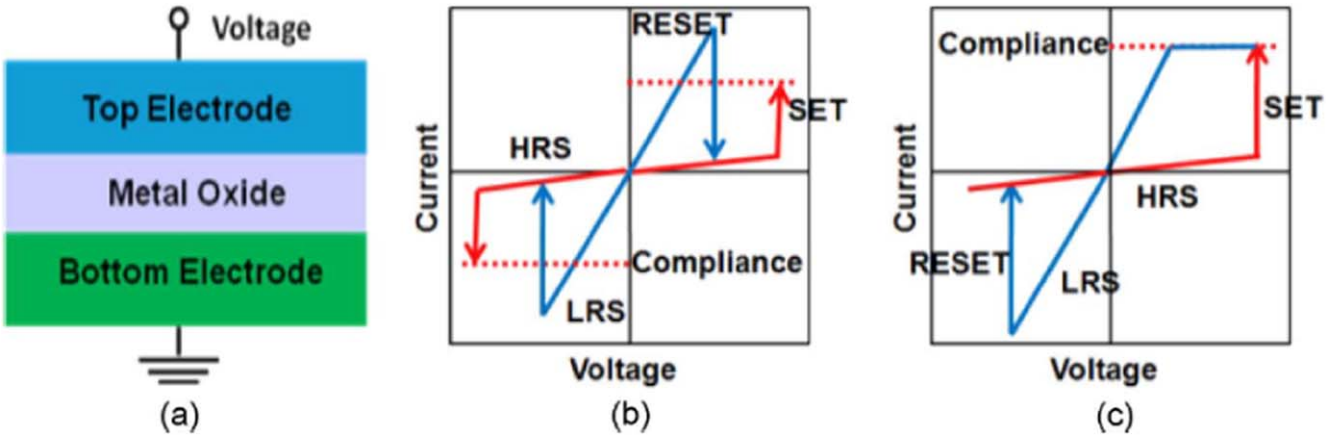

Figure 18. (a) Schematic of MIM structure for metal-oxide RRAM, and schematic of metal-oxide memory's *I–V* curves, showing two modes of operation: (b) unipolar and (c) bipolar. © [2012] IEEE. Reprinted, with permission, from [106].

implemented analog memory, synaptic plasticity and STDP functions using RRAM [108]. To mimic synaptic weight changes, a titanium oxide bilayer was applied to achieve interface resistance switching. Multi-level conductance control was realized by causing oxygen movement through two layers with different oxygen concentration of TiO_x/TiO_y. When a positive voltage is applied to the system, oxygen ions move from the TiO_y to the TiO_x layer, reducing the effective thickness of the layer and increasing the conductance. On the other hand, when a negative voltage is applied to the system, the effective thickness of the layer expands and the conductivity decreases.

Human memory is not permanent, but is strengthened into long-term memory (LTM) by repeated stimulus, and it easily disappears into short-term memory (STM). Forgetfulness is not always a disadvantage because it creates a space for storage for more important memories. Chang *et al* experimentally showed that the retention loss of nanoscale memristor devices is similar to memory loss in biological systems [109]. By stimulating the memristor with repeated voltage pulses, they implemented STM to LTM transition. The memristor device consisted of a W bottom electrode, Pd top electrode, and a WO_x film sandwiched in between. After repeated stimulation, there were sufficient oxygen vacancies in the switching layer, resulting in much improved retention along with the increase in conductance.

The filament formation process of RRAM is inherently unexpected and difficult to control due to the stochastic nature of filamentary switching. In particular, it can be a disadvantage in neuromorphic systems that require gradual current changes (i.e. weight updates). Yu *et al* reported RRAM with multilevel resistance states, which were obtained by varying the programming voltage amplitudes during the pulse cycling using multi-level metal oxide with the structure of $TiN/HfO_x/AlO_x/Pt$, $TiN/Ti/AlO_x/TiN$, and $TiN/TiO_x/HfO_x/TiO_x/HfO_x/Pt$ stacks, respectively [47, 52, 110]. They also reported high endurance (10^5 cycles) and low energy consumption per operation (sub-pJ). For pulse programming, no compliance current was enforced, which is possible because of the very short pulse width (~50 ns). Thus, excessive damage to the cell during the setup process was much reduced compared to the DC sweep case. Also, a stochastic compact model was developed to quantify the gradual resistance modulation and was applied to a large-scale artificial visual system simulation.

To implement multi-level conductance, Woo *et al* analyzed the response of identical pulses on a filamentary RRAM system with an $Al/HfO_2/Ti/TiN$ stack to implement the synapse function in neuromorphic computing systems [111]. The introduction of the AlO_x barrier layer was found to be advantageous for analog memory in enabling linear potentiation behavior as a function of the number of pulses due to the steadily expanded conductive filament. They also emphasized that using identical pulses can reduce the burden on the peripheral circuit. In the same group, the symmetric conductance change characteristics of the TiO_x-based RRAM were confirmed through the hybrid pulse scheme [112]. To achieve various conduction levels, interfacial resistance switching by redox reaction was adopted in the Mo/TiO_x stack. To improve the conductance variation symmetry, constant voltage and current pulses were applied during the enhancement and depression conditions, respectively. The mechanism analysis for the gradual reset phenomenon in Al_2O_3 RRAM was reported [113]. The reset was observed to be gradual when a significant number of vacancies were generated in the dielectric during the set event. In order to create a large number of vacancies in the dielectric, the forming step was divided into three parts (multi-step forming).

Since RRAM can be easily triggered by the stochastic nature of the oxygen vacancies, the effect of error on the system has been actively studied for neuromorphic systems. Zhao *et al* investigated the statistical behavior of read current noise and retention in a 1 kb filamentary analog RRAM array [114]. A standard multi-layer perceptron was used for the system and the MNIST data set was used for the recognition rate. They said that all conductance distributions follow a normal distribution and the standard deviation increases linearly with the square root of time. In addition, they emphasized that the retention effect is larger than the noise effect in terms of recognition rate error. Tosson *et al* provided a

modeling framework to compute the effect of soft errors on the system accuracy [115]. They said that the soft error of RRAM is caused by the diffusion of the oxygen vacancies, unbalanced programming pulses and manufacturing defects. Their simulation results showed that the system accuracy degraded from 91.6% to 43% due to the RRAM reliability soft errors. They also proposed methodologies for automatically detecting and fixing the degradation in the system accuracy. Using these methodologies, the system accuracy of their case-study system was increased from 43% to 91.6%.

Ambrogio *et al* introduced a new synaptic circuit consisting of a one-transistor/one-resistor structure [116, 117]. A fully connected neuromorphic network was simulated with on-chip unsupervised pattern learning and recognition. Only one transistor performs two functions for communication of the pre- to the postspike and weight update. With randomly alternated presentation of pattern and noise, on-chip learning was implemented. Prezioso *et al* demonstrated an STDP behavior that ensures self-adaptation of the average memristor conductance [118]. The synaptic weight change of the synapse was considered to be dependent not only on the presynaptic and postsynaptic signal, but the initial value of the synapse. However, they showed an STDP behavior that ensures self-adaptation of the average memristor conductance, making the plasticity stable. At least it is insensitive to the initial state of the device in a simple spike network.

For an energy- and cost-efficient neuromorphic computation hardware implementation, Wang *et al* proposed 3D synaptic architecture based on self-rectifying $Ta/TaO_x/TiO_2/Ti$ RRAM [119]. The analog synaptic plasticity was simulated using the compact models. A study involving more complex circuits and various neuromorphic applications was performed by Piccolboni *et al* [120]. A single synapse was composed by connecting one transistor and a stacked vertical RRAM (VRRAM) to realize analog-like conductance response with only two distinct resistive states of low resistance state (LRS) and high resistance state (HRS). A single-layer SNN was simulated for real-time auditory pattern extraction and CNN was demonstrated for the recognition of handwritten digits. Given a specific resistance distribution, the recognition rate varies with the number of stacked VRRAM cells, and they reported that more than 12 VRRAM cells are needed for a recognition rate of over 98%. Li *et al* developed a four-layer HfO_x-based 3D vertical RRAM with FinFET selector [121]. A system-level simulation was implemented with an unsupervised WTA visual system consisting of stochastic synapses and LIF neurons. They reported that the 3D architecture reduces interconnect RC effects and avoids long sneak leakage paths of the 2D architecture. Li *et al* introduced 3D RRAM structure using ternary levels that can overcome nonlinearities caused by premature 'analog' synapses [122]. They proposed a new operation scheme that combines the selected lines and the word-lines with an input vector and designs all bit-lines as weighted-sum outputs. They analyzed that the proposed 3D VRRAM implementation has a larger write margin for weighted sum/weight update, smaller latency and energy consumption for weight update compared to 2D structure.

3.2. CBRAM

CBRAM (atomic bridge) technology, which is also called programmable-metallization-cell memory, is known as one of the available memristor-based nonvolatile devices in constructing neuromorphic systems. Unlike the RRAM described previously, which is implemented using memristive materials, the conductance change of the CBRAM is achieved by using electrochemical characteristics [123]. To explain the basic operation of the CBRAM in more detail, it can be said that cations injected from one active electrode migrate to an electrolyte with a high ionic conductivity, as shown in figure 19(a). The formation of the active conductive pathway (metallic filament) between the two electrodes causes an LRS (set) of the device, while the insulation due to filament breakdown results in an HRS (reset) of the device, as described in figures 19(b) and (c). The formation and properties of the conductive filaments in CBRAM depend on the type of active conductivity. The implementation of multilevel resistance states caused by this analog and incremental change in conductance is commonly used to make synaptic plasticity in neuromorphic systems similar to other memristor-based devices. The ability to store multiple levels on a single storage device is one of the most important factors in emerging memory technology, and the multi-level capability can be achieved by controlling the programming current in the CBRAM device. In terms of scalability as an advantage of CBRAM, the threshold voltage and resistance parameter in the ON state are independent of the device size. However, the parameter determining resistance in the OFF state is dependent on the setting resolution, and the scalability limit is known to be in the range of 20 nm or less. Along with these properties, the advantages of the CBRAM in terms of its application to neuromorphic systems are fast speed (∼ns) and very low energy consumption (sub-pJ programming).

However, the CBRAM has a crucial disadvantage with the serious variability of the resistance modulation, which necessitates additional engineering requirements. Even the CBRAM can have poor retention characteristics compared to RRAM due to self-diffusion of the metal ion. In addition, the formation and destruction processes of the conductive filament are inherently abrupt and asymmetric. Although the required operating characteristics of the device may vary depending on the learning algorithm of the neuromorphic system, the incremental programmability maintaining a constant interval is well known as a factor that can have an effect on the learning accuracy.

Many studies on CBRAM have highlighted the merits of application to synaptic devices, and there have been many researches on evaluating the feasibility of using a single device. Jo *et al* demonstrated CBRAM consisting of a co-sputtered Ag and Si active layer with a proper Ag/Si mixture ratio gradient. This structure has Ag-rich and Ag-poor regions corresponding to high and low conductivities, respectively. The characteristics of the memristor device have shown that it can support important synaptic functions such as STDP [124]. Liu *et al* implemented an effective way to control the growth of the conductive filament using metal nanocrystals (Cu)

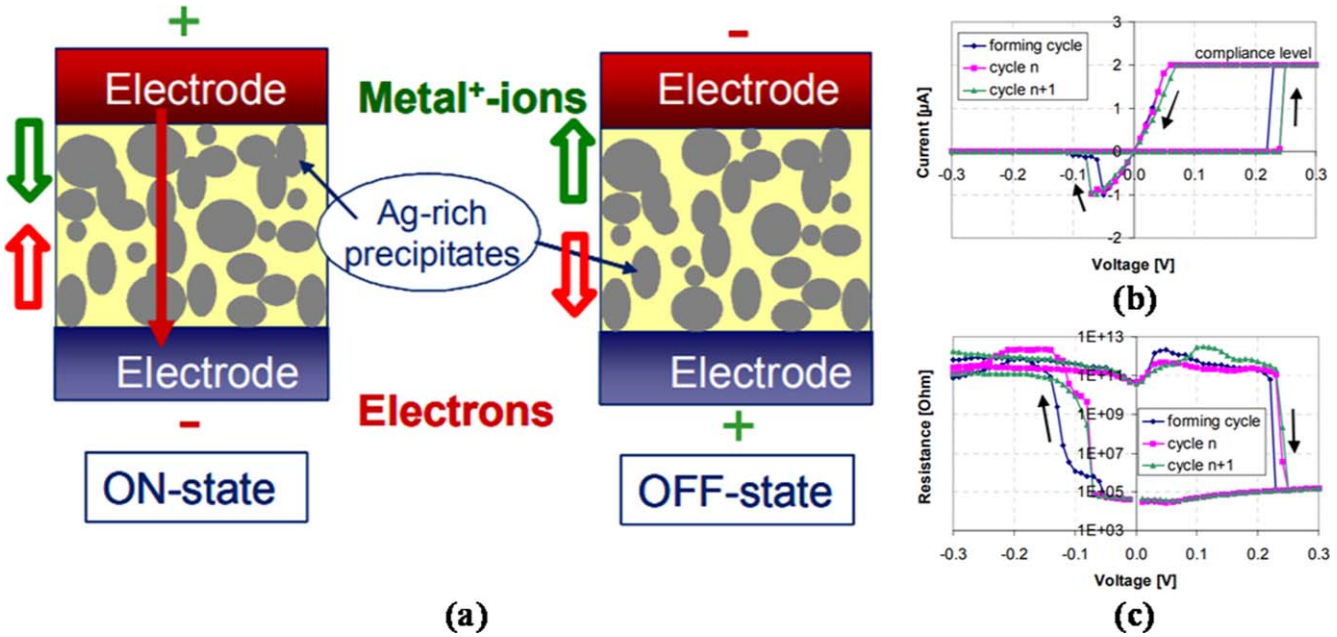

Figure 19. (a) Schematic illustration of the CBRAM switching mechanism. ON state is the LRS of the device caused by metallic filament. OFF state is the HRS of the device caused by filament breakdown. (b) Typical current–voltage characteristic of a CBRAM device. (c) Typical resistance–voltage characteristic of a CBRAM device. © [2005] IEEE. Reprinted, with permission, from [123].

covering the bottom electrode (Pt) in CBRAM, which has a $Ag/ZrO_2/Cu/Pt$ structure. Conductive filament can easily grow along the direction of metal nanocrystals, and their formation has the same path during repetitive switching cycles. This work demonstrated the possibility of controlling the conductive filament formation pathway using metal nanocrystals or other field-intensive initiators [125]. Ohno *et al* demonstrated the synaptic behavior of CBRAM using Ag_2S with experimental evidence of short-term plasticity and LTP characteristics, which are the key features of a biological synapse. Temporary changes in conductance over time were observed by less frequent input stimuli, while persistent enhancement was achieved by more frequent input stimuli [126]. Tsuruoka *et al* demonstrated that quantized conductance of CBRAM constructed with $Ag/Ta_2O_5/Pt$ can be achieved by applying the pulses having various amplitudes. LTP behavior occurs by applying consecutive input pulses at periodic intervals of different times. The realization of these devices is meaningful in that they are compatible with the CMOS manufacturing process [127, 128]. Roclin *et al* referred to the effects of sneak paths and parasitic metal line resistance in arrays of CBRAM operating as synapses for SNN. They showed that the structure of the crossbar array has a high energy consumption with high leakage during the transition of state, and an increased switching time due to voltage loss along the lines. Using the CBRAM in a digital manner, sensing the state of each CBRAM, can be a solution to mitigate this problem [129]. Nayak *et al* emulated the synaptic plasticity such as STM and LTM by using CBRAM using Cu_2S, and studied the sensitivity of the device towards the moisture and temperature. They demonstrated that the LTM is achieved much faster at elevated temperatures with shorter or fewer number of inputs. This work was the first temperature-dependence investigation of the CBRAM, and it is considered a study of synapses closer to the human brain [130].

In addition to these studies about the characteristics of a single device, there have also been modeling researches to determine the feasibility of CBRAM in high-level hierarchy. Yu *et al* developed a physical model reflecting the switching dynamics of CBRAM. The transient characteristics of CBRAM are in good agreement with the experimental result using Cu/SiO_2 and $Ag/Ge_{0.3}Se_{0.7}$, and the time-dependent switching process of CBRAM is quantified. The use of this model paves a more sophisticated way for mimicking synaptic functions using CBRAM and verifying the feasibility of STDP behavior in neuromorphic systems [131]. Related study shows that the SET transition in CBRAM that becomes stochastic under weak programming conditions was measured statistically and modeled for a WTA network. This study has shown that binary synapses can be used effectively in neuromorphic computing. In addition, the relaxation of constraints for designing continuous multi-level states implies the possibility of widening the choice of materials for synaptic devices [132].

Recent research trends in CBRAM have focused on implementing neuromorphic systems using various architectures rather than as a single device. Suri *et al* proposed the neuromorphic system with $Ag/GeS_2/W$ structured CBRAM as binary synapses, and used the STDP learning rule for unsupervised learning [133]. They showed various strategies for 1R and 1T-1R-based CBRAM configuration [134], and proposed a new methodology to design a low-power, low-footprint hardware architecture exploiting the intrinsic HRS variability of CBRAM

[135]. Gamrat *et al* implemented neuromorphic circuits suitable for embedded applications using CBRAM [136]. DeSalvo *et al* implemented a large-scale energy-efficient neuromorphic system with stochastic binary synapses based on CBRAM. CBRAM offers specific key features for applications as dense memory configuration and solving the field-programmable gate array (FPGA) leakage issue [137]. Mahalanabis *et al* demonstrated the possibility of tuning the ON state resistance of CBRAM with the analog STDP rule for neuromorphic applications [138].

Most of the above-mentioned studies show examples of CBRAM used as synaptic devices, since the advantages of CBRAM as a memristor are commonly highlighted in the neuromorphic system. However, there are also some studies that utilize the basic operating characteristics of CBRAM as neurons and not synaptic devices. Palma *et al* presented a methodology to design stochastic neurons using CBRAM. Unavoidable intrinsic variability on the time-to-set and off-state resistance of CBRAM were used to implement stochastic firing of neurons. An additional circuit and a novel self-programming technique for using CBRAM in I&F neurons was also proposed [139]. Jang *et al* designed a novel neuron circuit using a $Cu/Ti/Al_2O_3$-based CBRAM for HNN. This CBRAM-based neuron has the advantages of neuromorphic chip area and power aspects [140].

3.3. PCM

PCM, also known as PCRAM, is a type of nonvolatile RAM [105]. PCM uses the characteristics of large difference in electrical resistance of the phase of materials. The phase change states of the PCM can be divided into two states, which are amorphous and crystalline phases. The amorphous phase (HRS) has high electrical resistance (HRS). On the other hand, the crystalline phase (LRS) has a three to four order lower resistance value than the amorphous phase. PCMs should be prone to transition between the amorphous and crystalline phases, so they need to have characteristics of low melting temperature as well as crystallization temperature. In order to meet this demand, $Ge_2Sb_2Te_5$ (GST) is widely used as the PCM.

The structure of a typical PCM cell is shown in figure 20(a) [141]. A PCM cell consists of a top electrode, PCM, heater and bottom electrode. The electrical pulse passes through the PCM between the top electrode and heater, then the current is crowded at the heater to PCM contact. This crowded current makes heating power, which induces the programmable region such as the mushroom boundary. The phase change process of the PCM cell is divided into 'set' and 'reset'. The set is to put the PCM cell into LRS state. Likewise, the reset is put the PCM cell into HRS state. The set switching is a result of crystallization of the amorphous matrix in the programming region. Therefore, the set pulses must be able to heat above the crystalline temperature of PCMs, but below the melting temperature. The reset switching is a result of amorphization of the crystalline phase by melt quenching results. This is achieved when a large electrical current is applied to melt the central portion of the cell. Then, if the reset pulse is abruptly cut off, the melted material quenches into the amorphous phase, which has the HRS. Therefore, the reset pulses must be able to heat above the melting temperature and abrupt cut-off is required for temperature decrease. These pulse shapes are drawn in figure 20(b). The set pulse is similar, with program operation in conventional memory. The set pulse duration depends on the crystallization speed of the material. In addition, the reset pulse consumes large power because of the high melting point of PCMs. Therefore, the reset operation accounts for most of the power consumption in PCM applications.

PCM is a very attractive device in neuromorphic systems. First, the PCM device has a simpler fabrication process and simpler cell structure (two-terminal structure) than conventional memory devices (such as DRAM, NOR, and NAND flash) and general MOS devices. Therefore, it is competitive in neuromorphic systems where many synaptic arrays are needed (in the human brain, the number of synapses is more than 10^{15}). In addition, the PCM when used in synaptic devices, has advantages such as good retention, endurance and fast set speed. The retention ($\sim$10 year) and endurance ($>10^8$) [142] are important properties of synaptic devices.

On the other hand, there are limitations when PCM is used for neuromorphic systems. The biggest disadvantage is that the reset process is abrupt and difficult to control. In order to bring the device back to the amorphous state, the whole PCM must be sufficiently heated to melt and then should be cooled rapidly. Thus, the conductance is abruptly changed when a reset pulse is applied to the PCM device. This abrupt conductance change is not suitable for the implementation of an intermediate resistance in a synaptic device. In addition, in the process of applying the set pulse, heat is generated through the input signal to change the conductance and it is difficult to uniformly control the generated heat. Therefore, there is a variation between the device and the pulses, so the reliability of the device is reduced. In a neuromorphic system, the linear changes in conductance are more advantageous in learning accuracy. However, the changes in conductance of PCM is nonlinear. Moreover, a PCM device requires a selector device when used in a synapse array. The selector device is needed not only to select the synaptic device to which the voltage is applied, but to block unwanted current flow into the sneak path. A diode structure is widely used as such a selector device.

The research group using PCM devices in the ANN can be roughly divided into two groups that use the bio-inspired learning algorithm and software-based learning algorithm. These two groups focus on the NVM characteristics of PCM devices for synaptic devices in neuromorphic systems.

Suri *et al* used a PCM device for a synaptic device in a neuromorphic system based on the bio-inspired learning algorithm. They enhanced the performance of classic GST-based PCM devices by adding a thin HfO_2 interface layer [143]. The added HfO_2 layer lowers the set/reset current of the PCM device and increases the number of intermediate resistance states in the potentiation process. System power consumption is decreased to as low as 60 μW due to the increase in the number of conductance potentiating points, while individual synaptic programming power is decreased by more than 50% due to a

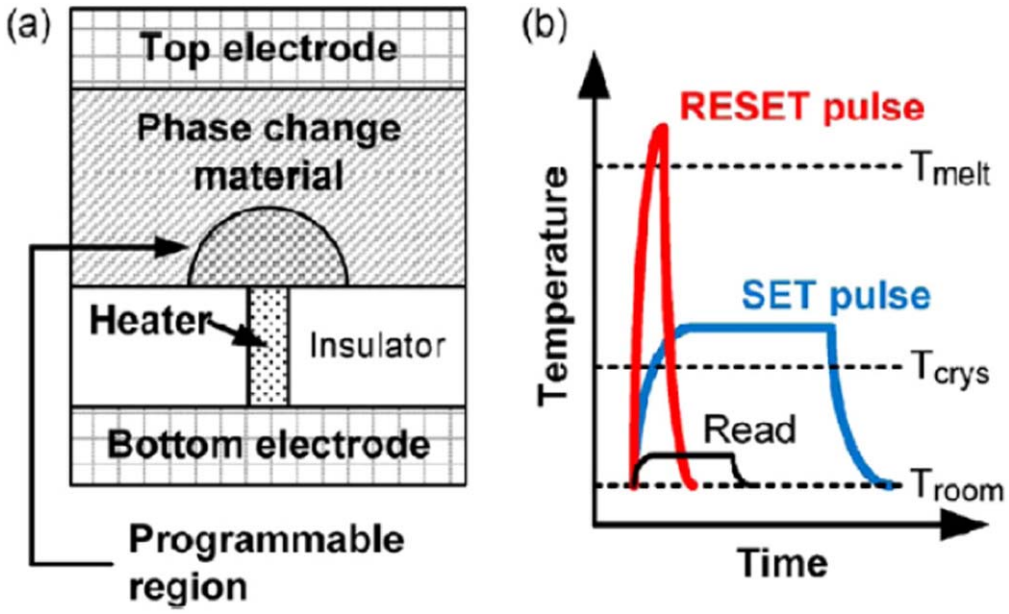

Figure 20. (a) Cross-sectional schematic of the conventional PCM cell. Current confined at the heater electrode and PCM contact results in a mushroom-shaped programmed region. (b) Temperature profile for reset and set switching, and read-out process. © [2010] IEEE. Reprinted, with permission, from [141].

decrease in the set and reset currents. In addition, Bichler *et al* used 2-PCM devices for mimicking one synapse in an SNN system. They used a simplified rule where LTP and LTD can both be produced with a single invariant crystallizing pulse [45]. Through the STDP, a biological learning algorithm, a real-world application of extracting complex patterns from recorded video data was implemented.

Kuzum *et al* implemented the symmetric and asymmetric STDP system with a single PCM device [144]. In that paper, characteristics of gradual change in conductance are implemented with different spike pulse schemes. The reset pulses of varying amplitude were used for the reset process, and staircase down pulses with varying amplitudes were used for a partial set process. Since the programming current of the PCM device is high, a short set/reset pulse is used for implementing a low-power system. Eryilmaz *et al* implemented a small-scale 10×10 crossbar array with selected PCM cells. They simulated a Hebbian STDP, which is similar to the biological brain [145]. The initial resistance variation was tolerated by lots of training epochs, but it consumes more energy.

Li *et al* investigated four different STDP forms, which are the antisymmetric Hebbian update with potentiation, anti-Hebbian update with potentiation, symmetric update with depression and symmetric update with potentiation, by applying programmed pre- and postsynaptic spiking pulse pairs in different time windows [146]. And then, they implemented those systems with a few PCM devices and simulations using square spike strategy (based on heat accumulation effect in PCM) [147].

Ambrogio *et al* presented a one-transistor/one-resistor (PCM cell at 45 nm node) synapse for neuromorphic systems [148]. This synapse is capable of STDP, and the learning results of single- and multi-pattern are demonstrated. By implementing a three-layer network, recognition accuracy is reached at 95.5% with 256 neurons (the error rate is 0.35%). The authors also proposed ways to reduce system power consumption by spiking communication.

Sidler *et al* enhanced the performance of the SNN by introducing an input encoding scheme that encodes the information from both the original and complementary pattern [46]. Compared with the case using conventional 2-PCM synapse, the number of devices are the same but the implementation is simpler, since there is no need for additional circuits for subtraction. So SNN can be realized more simply. However, the additional power consumption and area increase by peripheral circuits required to construct the complementary pattern inputs are not well addressed, making it difficult to directly make a comparison with the conventional 2-PCM synapse.

Ren *et al* improved the performance of PCM using O-Ti-Sb-Te (OTST) materials [149]. By reducing the oct-TCAM number in OTST, they could control the crystallization rate. Using this technique, linear conductance change, multi-bits ($\sim$8bits) and large on/off ratio ($\sim 10^2$) were achieved.

Burr *et al* used 2-PCM devices for one synaptic device based on the software-based learning algorithm for storing the synaptic weights, such as G^+, G^- [57]. In that paper, they explored the effects of non-ideal characteristics of synaptic devices on the recognition accuracy of neural networks. The analyzed non-ideal properties are nonlinearity, stochasticity, varying maxima, asymmetry between increasing/decreasing responses and nonresponsive devices at low or high conductance. Among these properties, the nonlinearity and asymmetry characteristics of synaptic devices cause a great loss of accuracy in neural networks. PCM devices have the nonlinear and asymmetric conductance change characteristics in the potentiation and depression process. Moreover, unidirectional changes in conductance degrade the accuracy of

the neural networks. Therefore, symmetric and bi-directional conductance changes are required.

3.4. Spin-based

Recently, spin-based memories have emerged as one of the candidates in NVM technology. The representative examples of the spin-based memories are STT and spin-obit torque (SOT) MRAM. The structure of conventional STT-MRAM is based on a magnetic tunnel junction (MTJ) consisting of a fixed magnetic layer, tunnel layer and free magnetic layer [150–153]. A resistance state of the STT-MRAM depends on the magnetization direction (spin-up and spin-down) of the free layer that is parallel or antiparallel to the fixed magnetic layer. When the magnetization direction of the free layer is antiparallel to the fixed magnetic layer, the MTJ is in an HRS. On the other hand, it is in an LRS when the direction of the free layer is parallel to the fixed magnetic layer. In addition, the resistance state of STT-MRAM is changed by flowing the current through the MTJ. When a positive current flows through the MTJ with the antiparallel state, the magnetization of the MTJ changes from antiparallel to parallel state due to the STT effect. Similarly, the magnetization changes from parallel to antiparallel state when a negative current flows through it. In the case of the SOT-MRAM, the device consists of a ferromagnet-heavy-metal heterostructure [154–158]. The resistance state of the SOT-MRAM also depends on the magnetization direction (spin-up and spin-down) of the magnetic layer. The magnetization direction is changed by SOT mechanism, which is generated by flowing the current through a heavy metal. The boundary between the regions of spin-up and spin-down means a domain wall in the magnetic layer. The conductance of the SOT-MRAM changes with the displacement of the domain wall. In addition, the magnetization of the magnetic layer is larger than the STT-MRAM due to SOT at the same current for changing the magnetization direction [154]. Therefore, the SOT-MRAM has a lower power consumption during the write operation than the STT-MRAM. However, the SOT-MRAM has three or four terminals to allow current to flow through the heavy metal to change the resistance state.

In terms of synaptic devices for the neuromorphic systems, spin-based memories display a relatively good performance in read and write operations with high speed (∼ns) and low energy consumption (∼pJ) compared to conventional RRAM and PCM [154, 155]. They also show excellent characteristics in terms of endurance ($>10^8$) [152]. However, the low on/off resistance ratio and the stochastic switching characteristics of spin-based memory devices are disadvantageous. Thus, spin-based memories have been mainly used as binary devices in memory applications. Several memory cell structures and materials have been studied, as shown in figure 21, to achieve a multi-level state at spin-memory-based synapses. In addition, spintronic neuromorphic systems have been researched to obtain a learning accuracy similar to that of RRAM and PCM-based neural network by using appropriate neural network algorithms to complement the stochastic switching characteristics of spin-based memories.

Many studies on STT-MRAM have highlighted the merits of its application to synaptic devices, and there have been many researches evaluating the feasibility of the neural network based on STT-MRAM. Zhang *et al* proposed a compound spintronic device consisting of multiple vertically stacked MTJs as a synaptic device to implement multiple resistance states [150]. These MTJs were composed of CoFeB/MgO/CoFeB thin films. The proposed compound spintronic device can achieve designable and stable multiple resistance states by interfacial and materials engineering from its components. Moreover, the proposed compound spintronic device was used to mimic neuron functionalities. All-spin artificial neural network was presented with the synaptic devices and neuron circuits based on the proposed compound spintronic device. The system-level simulations on the MNIST data set for handwritten digital recognition were performed and discussed with device variations. Lequeux *et al* suggested a spin-torque memristor as a synaptic device using the domain wall propagating in a magnetic track to implement multiple resistance states [151]. The magnetic stack consists of a synthetic anti-ferromagnet (CoPt/Ru/CoPt/Ta/FeB), tunnel barrier (MgO), magnetic free layer (FeB/Ta/FeB) and capping layer (MgO/Ta). The resistive switching mechanism of the proposed device was discussed. The conductance change due to the displacement of the magnetic domain wall by spin-torque was implemented for the synaptic weight change in neural computation. Vincent *et al* presented basic concepts relating to STT-MRAM behavior as learning-capable synaptic devices [152]. Three programming regimes (low, intermediate and high current) were identified and compared. Then, a neural network-inspired system was simulated to exploit the stochastic effect in performing unsupervised learning. The results demonstrated that the switching probabilities of the nano-devices did not need to be controlled perfectly. Sharad *et al* presented the ANN design using spin devices operated by pure spin-current injection for flipping a nano-magnet [153]. In order to implement synaptic weight and neuron functionalities, the conductance changes were investigated with the displacement of the domain wall in the magnetic layer. Then, CMOS-based inter-neuron communication was employed to realize network-level functionality using physics-based models of the spin devices.

Moreover, there have been many researches on evaluating the feasibility of the neural network based on SOT-MRAM. Sengupta *et al* proposed a ferromagnet-heavy-metal heterostructure based on SOT to implement the STDP. The proposed synaptic device consists of four terminals to decouple spike transmission and programming current paths. While the current for learning flows mainly through the heavy metal and the displacement of the magnetic domain wall changes, the spike current modulated by the MTJ conductance changes with the domain wall displacement. The performance and physical characteristics of the synaptic devices were discussed to implement the STDP [154]. Then, the physical mechanism for generating synaptic plasticity was investigated to implement the online programming of synapses based on the temporal information of spikes transmitted by spiking neurons [155]. It was also demonstrated that the magnetization dynamics of the MTJ can similarly implement the short-term plasticity and long-term

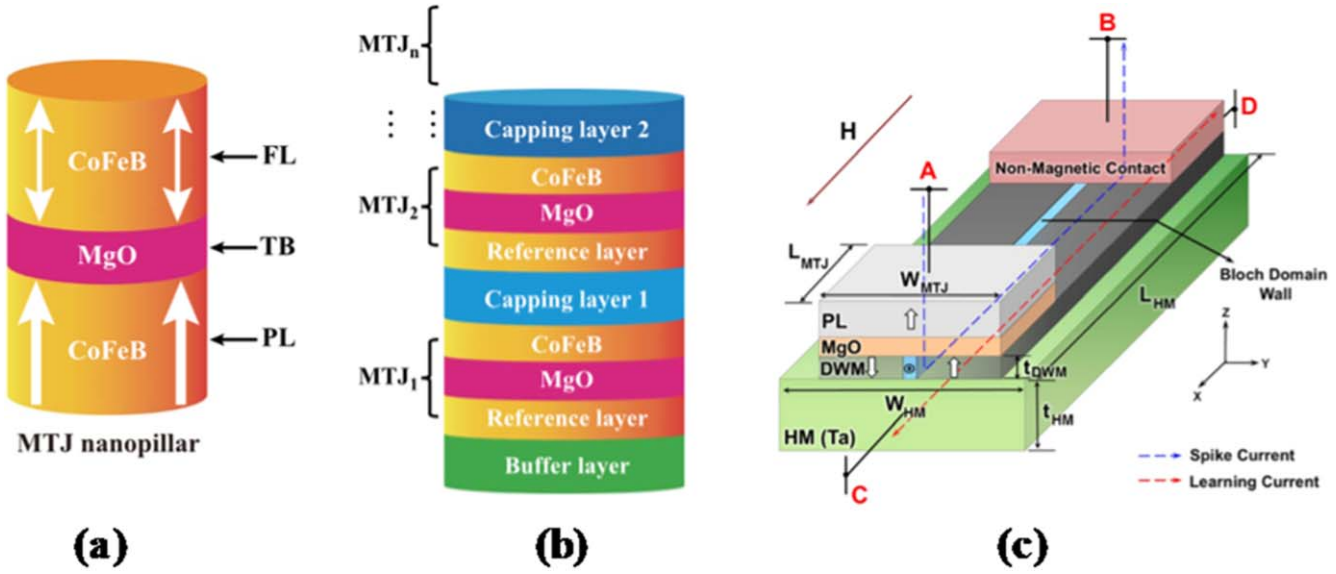

Figure 21. (a) Vertical structure schematic of an MTJ nanopillar composed of CoFeB/MgO/CoFeB thin films based on STT switching mechanism. Here, FL, TB and PL are short for free layer, tunnel barrier and pinned layer, respectively. (b) Vertical structure schematic of multiple vertically stacked MTJs to implement the multiple resistance states. © [2016] IEEE. Reprinted, with permission, from [150]. (c) 3D schematic of a four-terminal synaptic device based on SOT switching, which enables a conductance change using domain wall movement in a ferromagnetic layer. Reprinted from [154], with the permission of AIP Publishing.

plasticity of biological synapses [156]. The theoretical demonstration of short-term plasticity and long-term plasticity mechanisms in the MTJ was presented with the phenomenon of stabilizing the free layer in an antiparallel state according to the relative angle between the free layer and pinned layer. Finally, the low-power intelligent neuromorphic system with adaptive learning using short-term plasticity and long-term plasticity mechanisms was investigated. Srinivasan *et al* proposed a stochastic binary synapse composed of an MTJ and a heavy metal [157]. Synaptic plasticity was investigated by the stochastic switching of the MTJ conductance states, based on the temporal correlation between the spiking activities of the interconnecting neurons. The long-term and short-term stochastic synapses comprised two unique binary synaptic elements, respectively. The efficacy of the proposed synaptic configurations and stochastic learning algorithm was demonstrated in a trained SNN to classify handwritten digits in the MNIST data set. Borders *et al* demonstrated associative memory operations reminiscent of the brain using nonvolatile spintronics devices [158]. Anti-ferromagnet–ferromagnet bilayer-based Hall devices were used as the synaptic device, which showed analog-like SOT switching under zero magnetic fields. A neuromorphic system consisting of an FPGA and 36 SOT-MRAMs was designed. An effect of learning on the neuromorphic system using SOT-MRAMs was successfully confirmed for several 3 × 3-block patterns.

3.5. FET-based

To implement an HNN using the STDP algorithm, it is important to model the LTP and LTD functionalities as electrical elements in accordance with the spike firing sequence in actual biological synapses and neurons. The effective design of a synaptic device that can combine storage and computational capabilities is the core of the HNN implementation. Many studies have attempted to reproduce synaptic plasticity in electronic devices through a very large integrated circuit based on CMOS. Several circuits have been reported to simulate biological neurons. Recently, researches on constructing a synapse array using a memristor crossbar array have been actively conducted.

Synapses based on memristors have several advantages, but there are some areas that need improvement. First, memristor-based two-terminal synaptic devices can form a high-density synaptic device array [85], but most devices have a nonlinear conductance (G) response. Second, memristors are problematic in terms of device characteristic variation and reliability when deployed in large-scale crossbar arrays. The device characteristic variation of the memristor leads to the decrease of the recognition rate in the pattern recognition process of the HNN. One of the candidates for solving these problems is a CMOS FET. Carver Mead, a developer of the neuromorphic computing concept, proposed the first FET-based synapse in 1996 [159]. Mead and colleagues demonstrated a learning system that uses 2 × 2 synaptic arrays.

Studies on FET-based synaptic devices have been published and evolved into devices such as nanoparticle-organic FETs [160] and MemFlash [161]. Several groups have proposed FET-based synapse and neuron circuits using single or multiple carbon nanotubes [162].

In addition, an approach using a thin-film transistor (TFT)-type NOR flash memory cell with a half-covered floating gate as a synaptic device has been proposed, as shown in figure 22 [53]. The TFT-type NOR flash memory devices and device arrays have been fabricated using conventional CMOS fabrication processes. This structure allows the program/erase operation to be performed by changing the gate and source voltages so that the STDP characteristic of a

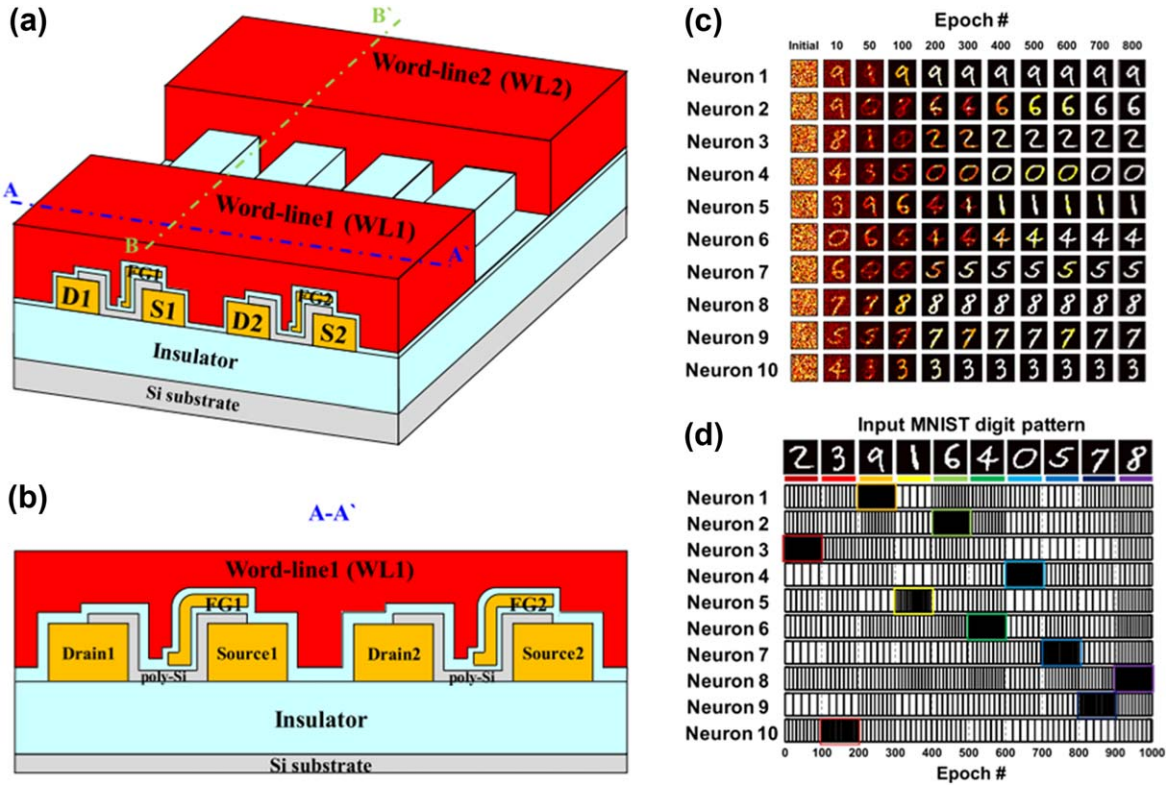

Figure 22. (a) Bird's eye view of a TFT-type NOR flash memory array and cross-sectional views cut in the (b) word-line direction. (c) Process of changing the weights of the synapses corresponding to each neuron. (d) Classification behavior of neurons when random digit patterns are applied after the multi-pattern learning process. © [2018] IEEE. Reprinted, with permission, from [53].

synapse can be implemented without any additional circuit configuration. In addition, word-lines and bit-lines are configured as the crossbar types and can be extended to large-scale synapse arrays. System-level simulation to classify the MNIST data set was successfully performed by adopting unsupervised learning using STDP in TFT-type NOR flash memory array. They used 28 × 28 MNIST handwritten digit patterns for learning and recognition processes.

A reconfigurable GSD was proposed as a new high-density and low-power synaptic device with near-linear *G*-response for HW-DNNs [85], as shown in figure 23. The proposed device is a GSD with a charge trap layer, which is fabricated using the unit processes of conventional Si CMOS technology. Since the Schottky junction between aluminum (Al) and poly-Si is located on the bottom gate (BG) covered with the $SiO_2/Si_3N_4/SiO_2$ stack, the effective Schottky barrier height is controlled by the BG bias or the amount of charge trapped in the Si_3N_4 layer. Note that the nitride layer (Si_3N_4) can store charges. Because the Schottky barrier is controlled under reverse bias, the reverse current is low enough (reverse current less than 12 $nA/\mu m$) to be used as a low-power synaptic device. The proposed device occupies a small area ($6F^2$), which is advantageous for implementing large-scale synaptic arrays. The Schottky reverse current has an exponential relationship with the effective Schottky barrier height that is related to the amount of stored charge, and the amount of stored charge is logarithmically proportional to the number of potentiation pulses. Since there are exponential and logarithmic relations canceling each other, a near-linear conductance response to the number of potentiation pulses can be obtained from the proposed device.

Another approach to mimic STM and LTM has been reported using two separated gates based on a FinFET structure [163]. One of the gates (G1) is used as a switching node, while the other gate (G2) is used as a memory node. Thanks to the electrically separated gates, the device can directly interact with the BP signal of the postsynaptic neuron circuit by G2 without any additional selection device and control circuit. Furthermore, STM and LTM are implemented and the transition between them depends on the interval between input pulses as in a biological system. The advantage of this scheme is that the synaptic transistors can directly interact with both pre- and postsynaptic neuron circuits.

A neuromorphic classifier using an embedded NOR flash memory array was proposed by Guo *et al* in 2017 [90]. They designed a 28 × 28 binary-input, ten-output, three-layer neuromorphic network using an embedded nonvolatile floating-gate cell array redesigned from commercially available 180 nm NOR flash memory. The main result reported in this paper is an experimental demonstration of a reproducible, stable and robust neuromorphic network that can classify the images of the standard MNIST dataset benchmark with high reliability, high speed and high energy efficiency.

A novel insulator-to-metal transition (IMT) FET is proposed for a synaptic device using an STDP algorithm by Stoliar *et al* [164]. They fabricated FET-type synaptic devices with $SrTiO_3$ channels, as shown in figure 24. This $SrTiO_3$ channel shows IMT characteristics due to the formation of a

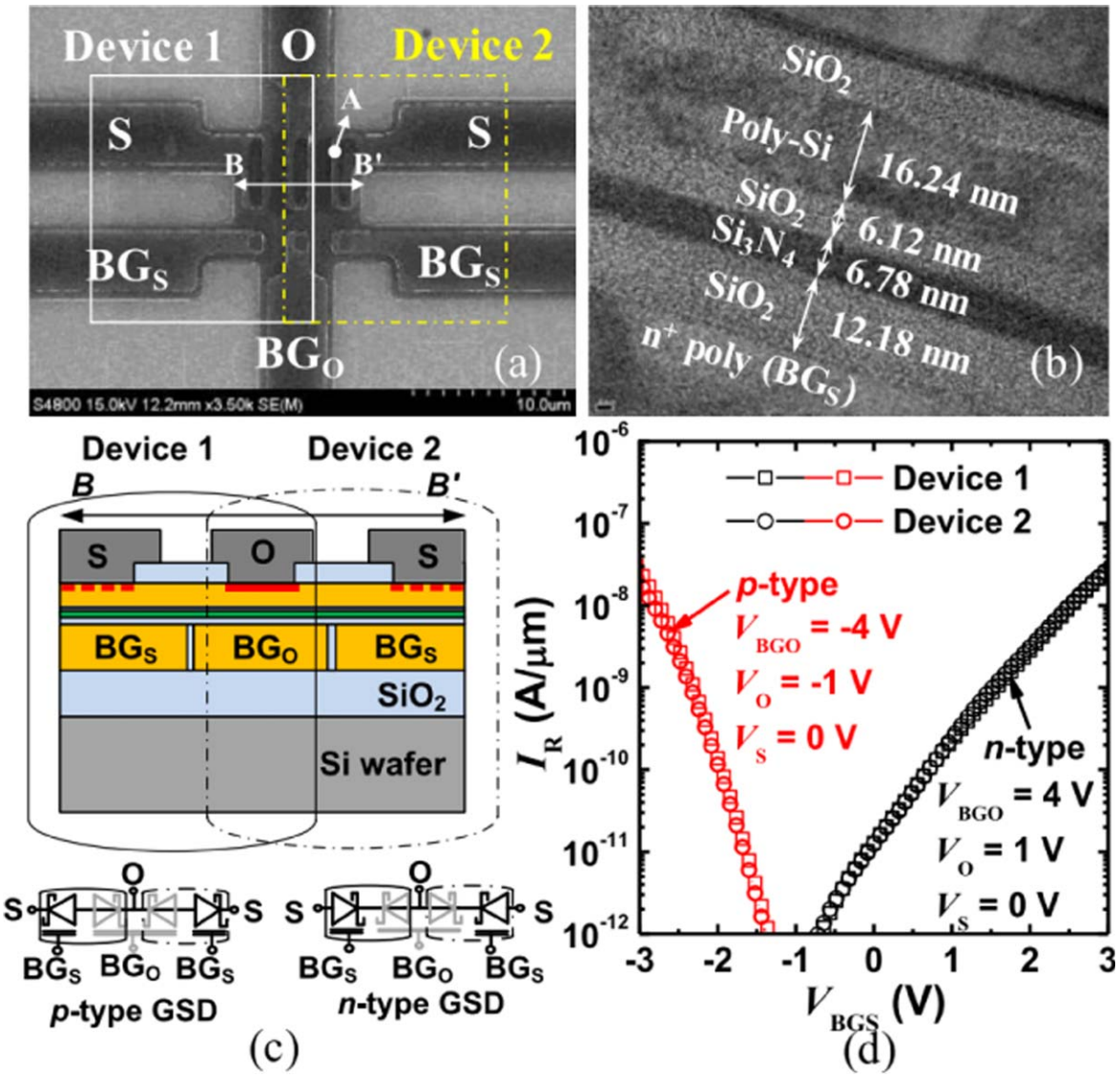

Figure 23. (a) Top SEM view. (b) Magnified cross-sectional TEM image at point A in (a). (c) Schematic cross-sectional view cut along the solid line B-B' in (a) and equivalent circuit diagram for p- and n-type GSDs. Dashed lines below the two Ss represent Schottky junctions, and the solid line below O represents an ohmic-like junction. (d) $I_R - V_{BGS}$ curves of p-/n-type GSD measured from two reconfigurable GSDs (Devices 1 and 2). V_{BGO} and V_O are negative for the p-type GSD and positive for the n-type GSD. © [2017] IEEE. Reprinted, with permission, from [85].

polar region in the bulk $SrTiO_3$. Furthermore, the power consumption of the IMT-FET device is much smaller than that of the resistive switching device.

Mulaosmanovic *et al* have reported a synaptic device based on a ferroelectric FET (FeFET) realized in a 28 nm high-k/metal gate technology [165]. Here, hafnium oxide is used as a ferroelectric material and gradual nonvolatile ferroelectric switching is used to mimic multi-level synaptic weights.

Another FeFET analog synaptic device has been proposed to accelerate DNN training [166]. The authors experimentally implemented an FeFET analog synaptic device using partial polarization switching to accelerate on-chip learning in DNN. The fabricated FeFET synaptic device exhibits symmetric 5-bit potentiation and depression characteristics. As a result, they showed an image recognition accuracy of 90% after training on the MNIST data set.

4. Conclusion

In this paper, we have reviewed neuromorphic technology for implementing ANNs. First, machine learning technology based on the current von Neumann architecture developed with the advancement of hardware accelerators has been introduced and its limitation including the very low energy efficiency have been investigated. To overcome these limitations, neuromorphic technology has been proposed, and discussed in two main directions. The first one is a neuromorphic technique using a bio-inspired learning algorithm

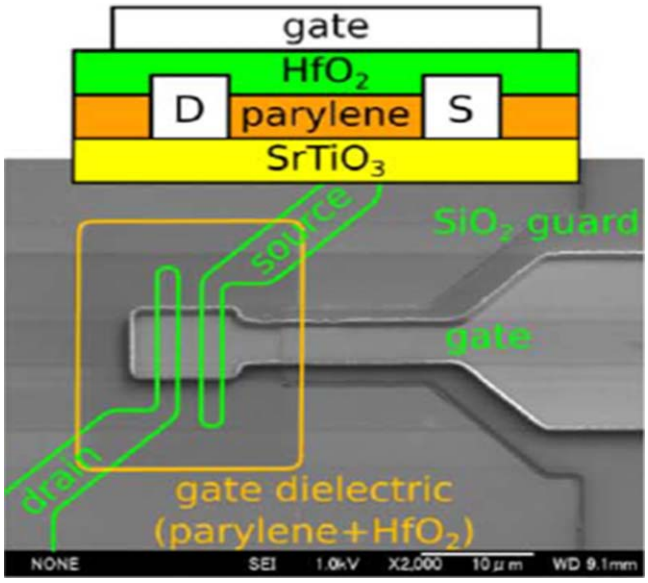

Figure 24. Cross-section (top) and SEM image (bottom) of our IMT-FET-based artificial synapse. © [2017] IEEE. Reprinted, with permission, from [164].

and the other is a neuromorphic technique using a software-based algorithm. We classified the types of neural networks applying various learning algorithms and summarized their characteristics, pros and cons. In detail, we analyzed the detailed classification of each algorithm and examined the characteristics of the synaptic devices required to implement the HNN using the corresponding algorithm. We also summarized the research progress using RRAM, CBRAM, PCM, spin-based memory and FET-based memory, which are representative emerging memory technologies used to implement synaptic devices and arrays. Through the analysis of developments in neuromorphic technology, which have been attracting the attention of researchers in various fields, we have presented a guideline that researchers should aim for. It is important to consider how to meet the requirements of memory devices for use as synaptic devices as well as how to configure the entire system of neural networks. Since the entire system of a neural network requires many additional ICs, the learning/inference method for the operation of a synaptic array must be carefully analyzed to match the compatibility with the peripheral ICs. At present, the application of neuromorphic research tends to be confined to visual pattern recognition. It is necessary to expand it to various application fields applicable to the real world such as stochastic computing, recognition of human voice and discrimination of atmospheric gas. If neuromorphic technology, which can present a new computing paradigm, can be successfully installed with various application possibilities, it will be able to open up a new horizon in the mobile artificial intelligence market by taking advantage of the low power consumption and high integration capability of the technology.

Acknowledgments

This work was partially supported by the MOTIE (Ministry of Trade, Industry & Energy) (10080583), the KSRC (Korea Semiconductor Research Consortium) support program for the development of the future semiconductor device, the KIST Institutional Program (Project No. 2E27810-18-P040), the National Research Foundation of Korea (NRF-2016M3A7B4909604) and the Brain Korea 21 Plus Project in 2018.

ORCID iDs

Chul-Heung Kim https://orcid.org/0000-0002-4419-7269
Suhwan Lim https://orcid.org/0000-0003-3578-5488
Sung Yun Woo https://orcid.org/0000-0002-0857-3183
Won-Mook Kang https://orcid.org/0000-0003-1812-3407
Young-Tak Seo https://orcid.org/0000-0003-3970-4876
Sung-Tae Lee https://orcid.org/0000-0002-7298-4382
Soochang Lee https://orcid.org/0000-0002-7554-143X
Dongseok Kwon https://orcid.org/0000-0001-7676-8938
Seongbin Oh https://orcid.org/0000-0003-4470-0554
Yoohyun Noh https://orcid.org/0000-0003-4150-8524
Hyeongsu Kim https://orcid.org/0000-0002-4157-5340
Jangsaeng Kim https://orcid.org/0000-0003-4519-135X
Jong-Ho Bae https://orcid.org/0000-0002-1786-7132
Jong-Ho Lee https://orcid.org/0000-0003-3559-9802

References

[1] Hinton G E and Salakhutdinov R R 2006 Reducing the dimensionality of data with neural networks *Science* **313** 504–7
[2] Mikolov T, Karafiát M, Burget L, Cernocký J and Khudanpur S 2010 Recurrent neural network based language model *Interspeech* **2** 045–8
[3] Srivastava N, Hinton G E, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58
[4] Krizhevsky A, Sutskever I and Hinton G E 2012 ImageNet classification with deep convolutional neural networks *Proc. Adv. Neural Information Processing Systems (NIPS)*
[5] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (https://doi.org/10.1109/cvpr.2015.7298594)
[6] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (https://doi.org/10.1109/cvpr.2016.90)
[7] Indiveri G and Liu S-C 2015 Memory and information processing in neuromorphic systems *Proc. IEEE* **103** 1379–97
[8] Poon C-S and Zhou K 2011 Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities *Front. Neurosci.* **22** 108
[9] Masquelier T and Thorpe S J 2007 Unsupervised learning of visual features through spike timing dependent plasticity *PLoS Comput. Biol.* **3** 247–57
[10] Burr G W *et al* 2015 Experimental demonstration and tolerancing of a large-scale neural network (165000

synapses) using phase-change memory as the synaptic weight element *IEEE Trans. Electron Devices* **62** 3498–507
[11] Merolla P A *et al* 2014 A million spiking-neuron integrated circuit with a scalable communication network and interface *Science* **345** 668–73
[12] Milo V, Pedretti G, Carboni R, Calderoni A, Ramaswamy N, Ambrogio S and Ielmini D 2016 Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2016.7838435)
[13] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proc. IEEE* **86** 2278–324
[14] Jouppi N P *et al* 2017 In-datacenter performance of a tensor processing unit *44th Annual Int. Symp. on Computer Architecture (ISCA)* (https://doi.org/10.1145/3079856.3080246)
[15] Chen Y-H, Krishna T, Emer J S and Sze V 2017 Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks *IEEE J. Solid-State Circuits* **52** 127–38
[16] Shafiee A, Nag A, Muralimanohar N, Balasubramonian R, Strachan J P, Hu M, Williams R S and Srikumar V 2016 ISAAC: a convolutional neural network accelerator with *in-situ* analog arithmetic in crossbars *43rd Annual Int. Symp. on Computer Architecture (ISCA)* (https://doi.org/10.1109/ISCA.2016.12)
[17] Chen Y *et al* 2014 DaDianNao: a machine-learning supercomputer *43rd Annual Int. Symp. on Microarchitecture (MICRO)*
[18] Chi P, Li S, Xu C, Zhang T, Zhao J, Liu Y, Wang Y and Xie Y 2016 PRIME: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory *43rd Annual Int. Symp. on Computer Architecture (ISCA)* (https://doi.org/10.1109/ISCA.2016.13)
[19] Cattani C and Pierro G 2013 On the fractal geometry of DNA by the binary image analysis *Bull. Math. Biol.* **75** 1544–70
[20] Diehl P U and Cook M 2015 Unsupervised learning of digit recognition using spike-timing-dependent plasticity *Front. Comput. Neurosci.* **9** 99
[21] Ĉadík M 2008 Perceptual evaluation of color-to-grayscale image conversions *Comput. Graph. Forum* 1745–54
[22] Ambrad M, Guo B, Martinez D and Bermak A 2008 A spiking neural network for gas discrimination using a tin oxide sensor array *IEEE Int. Symp. Elec. Des. Test. Appl.* (https://doi.org/10.1109/delta.2008.116)
[23] Wu S, Amari S I and Nakahara H 2002 Population coding and decoding in a neural field: a computational study *Neural Comput.* **14** 999–1026
[24] Vinje W E and Gallant J L 2000 Sparse coding and decorrelation in primary visual cortex during natural vision *Science* **287** 1273–6
[25] Adrian E D 1926 The impulses produced by sensory nerve endings *J. Physiol.* **61** 49–72
[26] Bienenstock E L, Cooper L N and Munro P W 1982 Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex *J. Neurosci.* **2** 32–48
[27] Indiveri G, Chicca E and Douglas R 2006 A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity *IEEE Trans. Neural Netw.* **17** 211–21
[28] O'Connor P, Neil D, Liu S C, Delbruck T and Pfeiffer M 2013 Real-time classification and sensor fusion with a spiking deep belief network *Front. Neurosci.* **7** 178
[29] Diehl P U, Neil D, Binas J, Cook M, Liu S C and Pfeiffer M 2015 Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing *Int. Joint Conf. on Neural Networks (IJCNN)* (https://doi.org/10.1109/ijcnn.2015.7280696)
[30] Querlioz D, Bichler O, Dollfus P and Gamrat C 2013 Immunity to device variations in a spiking neural network with memristive nanodevices *IEEE Trans. Nanotechnol.* **12** 288–95
[31] Wang Z *et al* 2017 Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing *Nat. Mater.* **16** 101
[32] Stein R B, Gossen E R and Jones K E 2005 Neuronal variability: noise or part of the signal? *Nat. Rev. Neurosci.* **6** 389–97
[33] Theunissen F and Miller J P 1995 Temporal encoding in nervous systems: a rigorous definition *J. Comput. Neurosci.* **2** 149–62
[34] Kaneko Y, Nishitani Y and Ueda M 2014 Ferroelectric artificial synapses for recognition of a multishaded image *IEEE Trans. Electron Devices* **61** 2827–8
[35] Sheik S, Pfeiffer M, Stefanini F and Indiveri G 2013 Spatio-temporal spike pattern classification in neuromorphic systems *Conf. Biomim. Biohybrid. Sys.* pp 262–73
[36] Querlioz D, Bichler O and Gamrat C 2011 Simulation of a memristor-based spiking neural network immune to device variations *Int. Joint Conf. on Neural Networks (IJCNN)* (https://doi.org/10.1109/ijcnn.2011.6033439)
[37] Kim H, Hwang S, Park J and Park B G 2017 Silicon synaptic transistor for hardware-based spiking neural network and neuromorphic system *Nanotechnology* **28** 405202
[38] Zeng Y, Devincentis K, Xiao Y, Ferdous Z I, Guo X, Yan Z and Berdichevsky Y 2018 A supervised STDP-based training algorithm for living neural networks *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP) (Calgary, AB)* pp 1154–8
[39] Querlioz D, Zhao W S, Dolfus P, Klein J O, Bichler O and Gamrat C 2012 Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches *IEEE/ACM Int. Sym. on Nanoscale Architectures (NANOARCH)* (https://doi.org/10.1145/2765491.2765528)
[40] Zamarreno-Ramos C, Camunas-Mesa L A, Perez-Carrasco J A, Masquelier T, Serrano-Gotarredona T and Linares-Barranco B 2011 On spike-timing-dependent plasticity, memristive devices, and building a self-learning visual cortex *Front. Neurosci.* **5** 26
[41] Bi G and Poo M 1998 Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type *J. Neurosci.* **18** 10464–72
[42] Almasi A D, Wozniak S, Cristea V, Leblebici Y and Engbersen T 2016 Review of advances in neural networks: Neural design technology stack *Neurocomputing* **174** 31–41
[43] Ambrogio S, Ciocchini N, Laudato M, Milo V, Pirovano A, Fantini P and Ielmini D 2016 Unsupervised learning by spike timing dependent plasticity in phase change memory (PCM) synapses *Front. Neurosci.* **10** 56
[44] Pedretti G, Bianchi S, Milo V, Calderoni A, Ramaswamy N and Ielmini D 2017 Modeling-based design of brain-inspired spiking neural networks with RRAM learning synapses *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2017.8268467)
[45] Bichler O, Suri M, Querlioz D, Vuillaume D, DeSalvo B and Gamrat C 2012 Visual pattern extraction using energy-efficient '2-PCM synapse' neuromorphic architecture *IEEE Trans. Electron Devices* **59** 2206–14
[46] Sidler S, Pantazi A, Wozniak S, Leblebici Y and Eleftheriou E 2017 Unsupervised learning using phase-change synapses and complementary patterns *Int. Conf. on Artificial Neural Networks (ICANN)* (https://doi.org/10.1007/978-3-319-68600-4_33)

[47] Yu S, Wu Y, Jeyasingh R, Kuzum D and Wong H S P 2011 An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation *IEEE Trans. Electron Devices* **58** 2729–37

[48] Adam G C, Hoskins B D, Prezioso M, Merrikh-Bayat F, Chakrabarti B and Strukov D B 2017 3D memristor crossbars for analog and neuromorphic computing applications *IEEE Trans. Electron Devices* **64** 312–8

[49] Chen P-Y, Lin B, Wang I-T, Hou T-H, Ye J, Vrudhula S, Seo J-S, Cao Y and Yu S 2015 Mitigating effects of non-ideal synaptic device characteristics for on-chip learning *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)* pp 194–9

[50] Yu S 2018 Neuro-inspired computing with emerging nonvolatile memory *Proc. IEEE* **106** 260–85

[51] Kim S, Lim M, Kim Y, Kim H-D and Choi S-J 2018 Impact of synaptic device variations on pattern recognition accuracy in a hardware neural network *Sci. Rep.* **8** 2638

[52] Yu S, Gao B, Fang Z, Yu H, Kang J and Wong H S P 2013 A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation *Adv. Mater.* **25** 1774–9

[53] Kim C-H, Lee S, Woo S Y, Kang W-M, Lim S, Bae J-H, Kim J and Lee J-H 2018 Demonstration of unsupervised learning with spike-timing-dependent plasticity using a TFT-type NOR flash memory array *IEEE Trans. Electron Devices* **65** 1774–80

[54] Choi H-S, Wee D-H, Kim H, Kim S, Ryoo K-C, Park B-G and Kim Y 2018 3D floating-gate synapse array with spike-time-dependent plasticity *IEEE Trans. Electron Devices* **65** 101–7

[55] Hinton G E 1986 Learning distributed representations of concepts *Proc. of the 8th Annual Conf. of the Cognitive Science Society*

[56] Hsu S K, Agarwal A, Anders M A, Mathew S K, Kaul H, Sheikh F and Krishnamurthy R K 2013 A 280 mV-to-1.1 V 256b reconfigurable SIMD vector permutation engine with 2-dimensional shuffle in 22 nm tri-gate CMOS *IEEE J. Solid-State Circuits* **48** 118–27

[57] Burr G W, Shelby R M, di Nolfo C, Jang J W, Shenoy R S, Narayanan P, Virwani K, Giacometti E U, Kurdi B and Hwang H 2014 Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses), using phase-change memory as the synaptic weight element *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2014.7047135)

[58] Liu B, Li H, Chen Y, Li X, Wu Q and Huang T 2015 Vortex: variation-aware training for memristor x-bar *ACM/EDAC/IEEE Design Automation Conf. (DAC)* (https://doi.org/10.1145/2744769.2744930)

[59] Prezioso M, Merrikh-Bayat F, Hoskins B D, Adam G C, Likharev K K and Strukov D B 2015 Training and operation of an integrated neuromorphic network based on metal-oxide memristors *Nature* **521** 61–4

[60] Hu M, Li H, Chen Y, Wu Q, Rose G S and Linderman R W 2014 Memristor crossbar-based neuromorphic computing system: a case study *IEEE Trans. Neural Netw. Learn. Syst.* **25** 1864–78

[61] Salakhutdinov R and Hinton G E 2007 Learning a nonlinear embedding by preserving class neighbourhood structure *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*

[62] Chen Y Y, Goux L, Clima S, Govoreanu B, Degraeve R, Sankar K G, Fantini A, Groeseneken G, Wouters D J and Jurczak M 2013 Endurance/retention trade-off on cap 1T1R bipolar RRAM *IEEE Trans. Electron Devices* **60** 1114–21

[63] Merrikh-Bayat F, Guo X, Klachko M, Prezioso M, Likharev K K and Strukov D B 2017 High-performance mixed-signal neurocomputing with nanoscale floating-gate memory cell arrays *IEEE Trans. Neural Netw. Learn. Syst.* pp 1–9

[64] Merrikh-Bayat F, Guo X, Om'mani H A, Do N, Likharev K K and Strukov D B 2015 Redesigning commercial floating-gate memory for analog computing applications *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (https://doi.org/10.1109/ISCAS.2015.7169048)

[65] Merrikh-Bayat F, Guo X, Klachko M, Do N, Likharev K and Strukov D B 2016 Model-based high-precision tuning of NOR flash memory cells for analog computing applications *Annual Device Research Conf. (DRC)* (https://doi.org/10.1109/drc.2016.7548449)

[66] Guo X, Merrikh-Bayat F, Prezioso M, Chen Y, Nguyen B, Do N and Strukov D B 2017 Temperature-insensitive analog vector-by-matrix multiplier based on 55 nm NOR flash memory cells *IEEE Custom Integrated Circuits Conf. (CICC)* (https://doi.org/10.1109/cicc.2017.7993628)

[67] Hasler J and Marr H Y 2013 Finding a roadmap to achieve large neuromorphic hardware systems *Front. Neurosci.* **7** 118

[68] Schlottmann C R and Hasler P E 2011 A highly dense, low power, programmable analog vector-matrix multiplier: the FPAA implementation *IEEE Trans. Emerg. Sel. Topics Circuits Syst.* **1** 403–11

[69] Yu S, Li Z, Chen P-Y, Wu H, Gao B, Wang D, Wu W and Qian H 2016 Binary neural network with 16 Mb RRAM macro chip for classification and online training *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2016.7838429)

[70] Hu M *et al* 2018 Memristor-based analog computation and neural network classification with a dot product engine *Adv. Mater.* **30** 1705914

[71] Gao L, Chen P-Y and Yu S 2016 Demonstration of convolution kernel operation on resistive cross-point array *IEEE Elecron Device Lett.* **37** 870–3

[72] Lim S, Bae J-H, Eum J-H, Lee S, Kim C-H, Kwon D, Park B-G and Lee J-H 2018 Adaptive learning rule for hardware-based deep neural networks using electronic synapse devices *Neural Comput. Applic.* 1–16

[73] Chang C-C *et al* 2017 Challenges and opportunities toward online training acceleration using RRAM-based hardware neural network *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2017.8268373)

[74] Binas J, Neil D, Indiveri G, Liu S-C and Pfeiffer M 2016 Precise deep neural network computation on imprecise low-power analog hardware arXiv:1606.07786

[75] Narayanan P, Sanches L L, Fumarola A, Shelby R M, Ambrogio S, Jang J, Hwang H, Leblebici Y and Bur G W 2017 Reducing circuit design complexity for neuromorphic machine learning systems based on non-volatile memory arrays *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (https://doi.org/10.1109/iscas.2017.8050988)

[76] Burr G W, Narayanan P, Shelby R M, Sidler S, Boybat I, di Nolfo C and Leblebici Y 2015 Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: Comparative performance analysis (accuracy, speed, and power) *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2015.7409625)

[77] Fumarola A, Narayanan P, Sanches L L, Sidler S, Jang J, moon K, Shelby R M, Hwang H and Bur G W 2016 Accelerating machine learning with non-volatile memory: exploring device and circuit tradeoffs *IEEE Int. Conf. on Rebooting Computing (ICRC)* (https://doi.org/10.1109/icrc.2016.7738684)

[78] Ambrogio S *et al* 2018 Equivalent-accuracy accelerated neural-network training using analogue memory *Nature* **558** 60–7

[79] Schiffmann W, Joost M and Werner R 1994 Optimization of the backpropagation algorithm for training multilayer perceptrons *Technical Report* University of Koblenz, Institute of Physics, Rheinau

[80] Nair M V and Dudek P 2015 Gradient-descent-based learning in memristive crossbar arrays *Int. Joint Conf. on Neural Networks (IJCNN)* (https://doi.org/10.1109/ijcnn.2015.7280658)

[81] Yao P *et al* 2017 Face classification using electronic synapses *Nat. Commun.* **8** 1–8

[82] Belhumeur P N, Hespanha J P and Kriegman D J 1997 Eigenfaces vs. fisherfaces: recognition using class specific linear projection *IEEE Trans. Pattern Anal. Mach. Intell.* **19** 711–20

[83] Wang Y-F, Lin Y-C, Wang I-T, Lin T-P and Hou T-H 2015 Characterization and modeling of nonfilamentary$Ta/TaO_x/TiO_2/Ti$ analog synaptic device *Sci. Rep.* **5** 10150

[84] Burr G W *et al* 2017 Neuromorphic computing using non-volatile memory *Adv. Phys.* X **2** 89–124

[85] Bae J-H, Lim S, Park B-G and Lee J-H 2017 High-density and near-linear synaptic device based on a reconfigurable gated Schottky diode *IEEE Electron Device Lett.* **38** 1153–6

[86] Lim S, Bae J-H, Eum J-H, Lee S, Kim C-H, Kwon D and Lee J-H 2018 Hardware-based neural networks using a gated Schottky diode as a synapse device *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (https://doi.org/10.1109/iscas.2018.8351152)

[87] Chang C-C, Chen P-C, Chou T, Wang I-T, Hudec B, Chang C-C, Tsai C-M, Chang T-S and Hou T-H 2018 Mitigating asymmetric nonlinear weight update effects in hardware neural network based on analog resistive synapse *IEEE J. Emerg. Sel. Topics Circuits Syst.* **8** 116–24

[88] moon K, Kwak M, Park J, Lee D and Hwang H 2017 Improved conductance linearity and conductance ratio of 1T2R synapse device for neuromorphic systems *IEEE Electron Device Lett.* **38** 1023–6

[89] Jang J-W, Park S, Jeng Y-H and Hwang H 2014 ReRAM-based synaptic device for neuromorphic computing *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (https://doi.org/10.1109/iscas.2014.6865320)

[90] Guo X, Merrikh-Bayat F, Bavandpour M, Klachko M, Mahmoodi M R, Prezioso M, Likharev K K and Strukov D B 2017 Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded NOR flash memory technology *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2017.8268341)

[91] Alibert F, Gao L, Hoskins B D and Strukov D B 2012 High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm *Nanotechnology* **23** 075201

[92] Wu H *et al* 2017 Device and circuit optimization of RRAM for neuromorphic computing *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2017.8268372)

[93] Nandakumar S R, Gallo M L, Boybat I, Rajendran B, Sebastian A and Eleftheriou E 2018 Mixed-precision architecture based on computational memory for training deep neural networks *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (https://doi.org/10.1109/iscas.2018.8351656)

[94] Boybat I, Gallo M L, Moraitis T, Leblebici Y, Sebastian A and Eleftheriou E 2017 Stochastic weight updates in phase-change memory-based synapses and their influence on artificial neural networks *IEEE 13th Conf. on PhD Research in Microelectronics and Electronics (PRIME)* (https://doi.org/10.1109/prime.2017.7974095)

[95] Cao Y, Chen Y and Khosla D 2015 Spiking deep convolutional neural networks for energy-efficient object recognition *Int. J. Comput. Vis.* **113** 54–66

[96] Rueckauer B, Lungu I-A, Hu Y, Pfeiffer M and Liu S-C 2017 Conversion of continuous-valued deep networks to efficient event-driven networks for image classification *Front. Neurosci.* **11** 682

[97] Mostafa H 2016 Supervised learning based on temporal coding in spiking neural networks *IEEE Trans. on Neural Networks and Learning Systems* vol 29, pp 3227–35

[98] Hunsberger E and Eliasmith C 2015 Spiking deep networks with LIF neurons arXiv:1510.08829

[99] Lee J H, Delbruck T and Pfeiffer M 2016 Training deep spiking neural networks using backpropagation *Front. Neurosci.* **10** 508

[100] Neftci E O, Augustine C, Paul S and Detorakis G 2017 Event-driven random backpropagation: enabling neuromorphic deep learning machines *Front. Neurosci.* **11** 324

[101] Nøkland A 2016 Direct feedback alignment provides learning in deep neural networks *Proc. Adv. Neural Information Processing Systems (NIPS)*

[102] Baldi P, Sadowski P and Lu Z 2017 Learning in the machine: the symmetries of the deep learning channel *Neural Netw.* **95** 110–4

[103] Lillicrap T P, Cownden D, Tweed D B and Akerman C J 2016 Random synaptic feedback weights support error backpropagation for deep learning *Nat. Commun.* **7** 13276

[104] Scellier B and Bengio Y 2017 Equilibrium propagation: bridging the gap between energy-based models and backpropagation *Front. Comput. Neurosci.* **11** 24

[105] Chua L O 2011 Resistance switching memories are memristors *Appl. Phys.* A **102** 765–83

[106] Wong H, Lee H, Yu S, Chen Y, Wu Y, Chen P, Lee B, Chen F and Tsai M 2012 Metal–oxide RRAM *Proc. IEEE* **100** 1951–70

[107] Yang J, Miao F, Pickett M, Ohlberg D, Stewart D, Lau C and Williams R 2009 The mechanism of electroforming of metal oxide memristive switches *Nanotechnology* **20** 215201

[108] Seo K *et al* 2011 Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device *Nanotechnology* **22** 254023

[109] Chang T, Jo S and Lu W 2011 Short-term memory to long-term memory transition in a nanoscale memristor *ACS Nano* **5** 7669–76

[110] Yu S, Gao B, Fang Z, Yu H, Kang J and Wong H S P 2012 A neuromorphic visual system using RRAM synaptic devices with sub-pj energy and tolerance to variability: experimental characterization and large-scale modeling *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2012.6479018)

[111] Woo J, moon K, Song J, Lee S, Kwak M, Park J and Hwang H 2016 Improved synaptic behavior under identical pulses using AlO_x/HfO_2 bilayer RRAM array for neuromorphic systems *IEEE Electron Device Lett.* **37** 994–7

[112] Park J, Kwak M, moon K, Woo J, Lee D and Hwang H 2016 TiO_x-based RRAM synapse with 64-levels of conductance and symmetric conductance change by adopting a hybrid pulse scheme for neuromorphic computing *IEEE Electron Device Lett.* **37** 1559–62

[113] Sarkar B, Lee B and Misra V 2015 Understanding the gradual reset in $Pt/Al_2O_3/Ni$ RRAM for synaptic applications *Semicond. Sci. Technol.* **30** 105014

[114] Zhao M *et al* 2017 Investigation of statistical retention of filamentary analog RRAM for neuromorphic computing *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2017.8268522)

[115] Tosson A, Yu S, Anis M and Wei L 2017 *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **25** 3125–37

[116] Ambrogio S, Balatti S, Milo V, Carboni R, Wang Z, Calderoni A, Ramaswamy N and Ielmini D 2016 Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM *IEEE Trans. Electron Devices* **63** 1508–15

[117] Pedretti G, Milo V, Ambrogio S, Carboni R, Bianchi S, Calderoni A, Ramaswamy N, Spinelli A and Ielmini D 2018 Stochastic learning in neuromorphic hardware via spike timing dependent plasticity with RRAM synapses *IEEE J. Emerg. Sel. Topics Circuits Syst.* **8** 77–85

[118] Prezioso M, Merrikh-Bayat F, Hoskins B D, Likharev K K and Strukov D B 2016 Self-adaptive spike-time-dependent plasticity of metal-oxide memristors *Sci. Rep.* **6** 21331

[119] Wang I, Lin Y, Wang Y, Hsu C and Hou T 2014 3D synaptic architecture with ultralow sub-10 fJ energy per spike for neuromorphic computation *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2014.7047127)

[120] Piccolboni G *et al* 2015 Investigation of the potentialities of Vertical Resistive RAM (VRRAM) for neuromorphic applications *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2015.7409717)

[121] Li H *et al* 2016 Four-layer 3D vertical RRAM integrated with FinFET as a versatile computing unit for brain-inspired cognitive information processing *Symp. on VLSI Technology (VLSIT)* (https://doi.org/10.1109/vlsit.2016.7573431)

[122] Li Z, Chen P, Xu H and Yu S 2017 Design of ternary neural network with 3D vertical RRAM array *IEEE Trans. Electron Devices* **64** 2721–7

[123] Kund M, Beitel G, Pinnow C-U, Rohr T, Schumann J, Symanczyk R, Ufert K and Muller G 2005 Conductive bridging RAM (CBRAM): an emerging non-volatile memory technology scalable to sub 20 nm *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2005.1609463)

[124] Jo S H, Chang T, Ebong I, Bhadviyq B B, Mazumder P and Lu W 2010 Nanoscale memristor device as synapse in neuromorphic systems *Nano Lett.* **10** 1297–301

[125] Liu Q, Long S, Lv H, Wang W, Niu J, Huo Z, Chen J and Liu M 2010 Controllable growth of nanoscale conductive filaments in solid-electrolyte-based ReRAM by using a metal nanocrystal covered bottom electrode *ACS Nano.* **4** 6162–8

[126] Ohno T, Hasegawa T, Tsuruoka T, Terabe K, Gimzewski J K and Aono M 2011 Short-term plasticity and long-term potentiation mimicked in single inorganic synapses *Nat. Mater.* **10** 591–5

[127] Tsuruoka T, Hasegawa T, Terabe K and Aono M 2012 Conductance quantization and synaptic behavior in a Ta_2O_5-based atomic switch *Nanotechnology* **23** 435705

[128] Tsuruoka T, Hasegawa T and Aono M 2014 Synaptic plasticity and memristive behavior operated by atomic switches *14th Int. Workshop on Cellular Nanoscale Networks and their Applications (CNNA)* (https://doi.org/10.1109/cnna.2014.6888615)

[129] Roclin D, Bichler O, Gamrat C and Klein J-O 2014 Sneak paths effects in CBRAM memristive devices arrays for spiking neural networks *IEEE/ACM Int. Symp. on Nanoscale Architectures (NANOARC)* (https://doi.org/10.1145/2770287.2770291)

[130] Nayak A, Ohno T, Tsuruoka T, Terabe K, Hasegawa T, Gimzewski J K and Aono M 2012 Controlling the synaptic plasticity of a Cu_2S gap-type atomic switch *Adv. Funct. Mater.* **22** 3606–13

[131] Yu S and Wong H S P 2010 Modeling the switching dynamics of programmable-metallization-cell (PMC) memory and its application as synapse device for a neuromorphic computation system *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2010.5703410)

[132] Yu S, Gao B, Fang Z, Yu H, Kang J and Wong H S P 2013 Stochastic learning in oxide binary synaptic device for neuromorphic computing *Front. Neurosci.* **7** 186

[133] Suri M, Bichler O, Querlioz D, Palma G, Vianello E, Vuillaume D, Gamrat C and DeSalvo B 2012 CBRAM devices as binary synapses for low-power stochastic neuromorphic systems: auditory (Cochlea) and visual (Retina) cognitive processing applications *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2012.6479017)

[134] Suri M, Querlioz D, Bichler O, Palma G, Vianello E, Vuillaume D, Gamrat C and DeSalvo B 2013 Bio-inspired stochastic computing using binary CBRAM synapses *IEEE Trans. on Electron Devices* **60** 2402–9

[135] Suri M and Parmar V 2015 Exploiting intrinsic variability of filamentary resistive memory for extreme learning machine architectures *IEEE Trans. Nanotechnol.* **14** 963–8

[136] Gamrat C, Bichleer O and Roclin D 2015 Memristive based device arrays combined with spike based coding can enable efficient implementations of embedded neuromorphic circuits *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2015.7409626)

[137] DeSalvo B, Vianello E, Thomas O, Clermidy F, Bichler O, Gamrat C and Perniola L 2015 Emerging resistive memories for low power embedded applications and neuromorphic systems *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (https://doi.org/10.1109/iscas.2015.7169340)

[138] Mahalanabis D, Sivaraj M, Chen W, Shah S, Barnaby H J, Kozicki M N, Blain Christen J and Vrudhula S 2016 Demonstration of spike timing dependent plasticity in CBRAM devices with silicon neurons *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (https://doi.org/10.1109/iscas.2016.7539047)

[139] Palma G, Suri M, Querlioz D, Vianello E and De Salvo B 2013 Stochastic neuron design using conductive bridge RAM *IEEE/ACM Int. Symp. on Nanoscale Architectures (NANOARC)* (https://doi.org/10.1109/nanoarch.2013.6623051)

[140] Jang J-W, Attarimashalkoubeh B, Prakash A, Hwang H and Jeong Y-H 2016 Scalable neuron circuit using conductive-bridge RAM for pattern reconstructions *IEEE Trans. Electron Devices* **63** 2610–3

[141] Wong H S P, Kim S B, Liang J, Reifenberg J P, Rajendran B, Asheghi M and Goodson K E 2010 Phase change memory *Proc. IEEE* **98** 2201

[142] Burr G W *et al* 2016 Recent progress in phase-change memory technology *IEEE J. Emer. Sel. Topics Circuits Syst.* **6** 146–62

[143] Suri M, Bichler O, Hubert Q, Perniola L, Sousa V, Jahan C, Vuillaume D, Gamrat C and Desalvo B 2012 Interface engineering of PCM for improved synaptic performance in neuromorphic systems *IEEE Int. Memory Workshop (IMW)* (https://doi.org/10.1109/imw.2012.6213674)

[144] Kuzum D, Jeyasingh R D and Wong H S 2011 Energy efficient programming of nanoelectronic synaptic devices for large-scale implementation of associative and temporal sequence learning *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2011.6131643)

[145] Eryilmaz S B, Kuzum D, Jeyasingh G D, Kim S B, BrightSky M, Lam C and Wong H S P 2013 Experimental demonstration of array-level learning with phase change synaptic devices *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2013.6724691)

[146] Li Y, Zhong Y P, Xu L, Zhang J J, Xu X H, Sun H J and Miao X S 2013 Ultrafast synaptic events in a chalcogenide memristor *Sci. Rep.* **3** 1619

[147] Zhong Y P, Li Y, Xu L and Miao X 2015 Simple square pulses for implementing spike-timing-dependent plasticity in phase-change memory *Phys. Status Solidi Rapid Res. Lett.* **9** 414

[148] Ambrogio S, Ciocchini N, Laudato M, Milo V, Pirovano A, Fantini P and Ielmini D 2016 Unsupervised learning by spike timing dependent plasticity in phase change memory (PCM) synapse *Front. Neurosci.* **10** 56

[149] Ren K, Li R, Chen X, Wang Y, Shen J, Xia M, Lv S, Ji Z and Song Z 2018 Controllable SET process in O-Ti-Sb-Te based

phase change memory for synaptic application *Appl. Phys. Lett.* **112** 073106
[150] Zhang D, Zeng L, Cao K, Wang M, Peng S, Zhang Y, Zhang Y, Klein J-O, Wang Y and Zhao W S 2016 All spin artificial neural networks based on compound spintronic synapse and neuron *IEEE Trans. Biomed. Circuits Syst.* **10** 828–36
[151] Lequeux S, Sampaio J, Cros V, Yakushiji K, Fukushima A, Matsumoto R, Kubota H, Yuasa S and Grollier J 2016 A magnetic synapse: multilevel spin-torque memristor with perpendicular anisotropy *Sci. Rep.* **6** 31510
[152] Vincent A F, Larroque J, Zhao W S, Romdhane N B, Bichler O, Gamrat C, Klein J-O, Galdin-Retailleau S and Querlioz D 2014 Spin-transfer torque magnetic memory as a stochastic memristive synapse *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (https://doi.org/10.1109/iscas.2014.6865325)
[153] Sharad M, Augustine C, Panagopoulos G and Roy K 2012 Spin-based neuron model with domain-wall magnets as synapse *IEEE Trans. Nanotechnol.* **11** 843–53
[154] Sengupta A, Azim Z A, Fong X and Roy K 2015 Spin–orbit torque induced spike-timing dependent plasticity *Appl. Phys. Lett.* **106** 093704
[155] Sengupta A, Banerjee A and Roy K 2016 Hybrid spintronic-CMOS spiking neural network with on-chip learning: devices, circuits, and systems *Phys. Rev. Appl.* **6** 064003
[156] Sengupta A and Roy K 2016 Short-term plasticity and long-term potentiation in magnetic tunnel junctions: towards volatile synapses *Phys. Rev. Appl.* **5** 024012
[157] Srinivasan G, Sengupta A and Roy K 2016 Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning *Sci. Rep.* **6** 29545
[158] Borders W A, Akima H, Fukami S, Moriya S, Kurihara S, Horio Y, Sato S and Ohno H 2017 Analogue spin–orbit torque device for artificial-neural-network-based associative memory operation *Appl. Phys. Express* **10** 013007
[159] Diorio C, Hasler P, Minch B A and Mead C A 1996 A single-transistor silicon synapse *IEEE Trans. Electron Devices* **43** 1972–80
[160] Alibart F, Pleutin S, Bichler O, Gamrat C, Serrano-Gotarredona T, Linares-Barraco B and Vuillaume D 2012 A memristive nanoparticle/organic hybrid synapstor for neuro-inspired computing *Adv. Funct. Mater.* **22** 609–16
[161] Riggert C, Ziegler M, Schroeder D, Krautschneider W H and Kohlstedt H 2014 MemFlash device: floating gate transistors as memristive devices for neuromorphic computing *Semicond. Sci. Technol.* **29** 104011
[162] Kuzum D, Yu S and Wong H S 2013 Synaptic electronics: materials, devices and applications *Nanotechnology* **24** 382001
[163] Kim H, Park J, Kwon M-W, Lee J-H and Park B-G 2016 Silicon-based floating-body synaptic transistor with frequency-dependent short- and long-term memories *IEEE Electron Device Lett.* **37** 249–52
[164] Stoliar P, Schulman A, Kitoh A, Sawa A and Inoue I H 2017 STDP synapse with outstanding stability based on a novel insulator-to-metal transition FET *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2017.8268506)
[165] Mulaosmanovic H, Ocker J, Müller S, Noack M, Müller J, Polakowski P, Mikolajick T and Slesazeck S 2017 Novel ferroelectric FET based synapse for neuromorphic systems *Symp. on VLSI Technology (VLSIT)* (https://doi.org/10.23919/vlsit.2017.7998165)
[166] Jerry M, Chen P-Y, Zhang J, Sharma P, Ni K, Yu S and Datta S 2017 Ferroelectric FET analog synapse for acceleration of deep neural network training *IEEE Int. Electron Devices Meeting (IEDM)* (https://doi.org/10.1109/iedm.2017.8268338)