

ACCEPTED MANUSCRIPT

## memristor devices for neural networks

To cite this article before publication: Hongsik Jeong *et al* 2018 *J. Phys. D: Appl. Phys.* in press <https://doi.org/10.1088/1361-6463/aae223>

### Manuscript version: Accepted Manuscript

Accepted Manuscript is “the version of the article accepted for publication including all changes made as a result of the peer review process, and which may also include the addition to the article by IOP Publishing of a header, an article ID, a cover sheet and/or an ‘Accepted Manuscript’ watermark, but excluding any other editing, typesetting or other changes made by IOP Publishing and/or its licensors”

This Accepted Manuscript is © 2018 IOP Publishing Ltd.

During the embargo period (the 12 month period from the publication of the Version of Record of this article), the Accepted Manuscript is fully protected by copyright and cannot be reused or reposted elsewhere.

As the Version of Record of this article is going to be / has been published on a subscription basis, this Accepted Manuscript is available for reuse under a CC BY-NC-ND 3.0 licence after the 12 month embargo period.

After the embargo period, everyone is permitted to use copy and redistribute this article for non-commercial purposes only, provided that they adhere to all the terms of the licence <https://creativecommons.org/licenses/by-nc-nd/3.0>

Although reasonable endeavours have been taken to obtain all necessary permissions from third parties to include their copyrighted content within this article, their full citation and copyright line may not be present in this Accepted Manuscript version. Before using any content from this article, please refer to the Version of Record on IOPscience once published for full citation and copyright details, as permissions will likely be required. All third party content is fully copyright protected, unless specifically stated otherwise in the figure caption in the Version of Record.

View the [article online](#) for updates and enhancements.

# Memristor Devices for Neural Networks

Hongsik Jeong and Luping Shi<sup>\*,†</sup>

Department of E.E. and CBICR, Tsinghua University, Beijing 100084, China

<sup>†</sup>Department of Precision Instrument. and CBICR, Tsinghua University, Beijing 100084, China

## Abstract

Neural network technologies have taken center stage owing to their powerful computing capability to support deep learning in artificial intelligence. However, conventional synaptic devices such as SRAM and DRAM are not satisfactory solutions for neural networks. Recently several types of memristor devices have become popular alternatives because of their outstanding characteristics such as scalability, high performance, and non-volatility. To understand the characteristics of memristors, a comparison among memristors has been made, considering both maturity and performance. MRAM, PRAM, and RRAM among proposed memristors are good candidates as synaptic devices for weight storage and matrix-vector multiplication required in artificial neural networks (ANNs). Moreover, these devices play key roles as synaptic devices in research for bio-plausible spiking neural networks (SNNs) because their distinctive switching properties are well matched to emulating synaptic and neuron functions of biological neural networks. In this paper we review motivation, advantage, technology, and applications of memristor devices for neural networks from practical approaches of ANNs to futuristic research of SNNs, considering the current status of memristor technology.

## 1. Introduction

People have believed that nature involves the beauty of symmetry emerging from the same origin. In this manner, the conventional electromagnetic passive circuit elements which are resistor, capacitor, and inductor are not sufficient to explain the symmetry of circuit theory connecting all electromagnetic properties. To compensate for this imperfection, a fourth element, called a memristor, was predicted in 1971 [1]. The memristor was originally defined as a component that connected charge and magnetic flux [1,2], and had a pinched-hysteresis loop I-V relationship whose size is frequency dependent [3]. Over 40 years the existence of the memristor remained theoretical, but finally, experimental evidence involving a  $\text{TiO}_2/\text{TiO}_{2-x}$  RRAM was observed and announced in 2008 [4]. Since the first evidence of memristor existence in RRAM, diverse types of memristor devices such as MRAM, PRAM, and FeRAM have been reported [5-9]. Even though their potential as memristor devices was later discovered, they were originally developed as non-volatile memory devices. The earliest research and basic physical mechanism of these devices occurred before 1970's and some prototypes at the industrial level were introduced in the early 2000's. These memristor devices were mainly considered for memory device applications owing to their outstanding characteristics such as non-volatility, high density, fast speed, and high reliability [10-14]. Moreover, some of the memristor devices also have been considered good candidates for neural network application because of their feasible characteristics as neuromorphic devices [15-18]. Neural network technologies have been studied extensively from fundamental research to real applications. Inspiration from biological functions of the brain for more efficient computing (the collaboration with fundamental research related to neuroscience) has played an important role toward enhancing neural network technologies [19-21]. Besides fundamental research, practical neural network technologies have been developed to realize artificial intelligence applications such as autonomous driving vehicles, robotics, image processing, translation, medical diagnosis, finance, advanced computing, pattern recognition, and so on [22-30]. To support these diverse artificial intelligence (AI) applications, various neural networks have been inspired from neuroscience and have evolved in a

different manner than conventional Von-Neumann computing [31].

Since the single layer neural networks such as the McCulloch-Pitts neuron and Perceptron, which are referred to as 1<sup>st</sup> generation neural network models, neural network technologies have advanced drastically [32]. Artificial neural networks(ANNs) having computational units that apply an activation function with a continuous set of possible output values to a weighted sum of inputs, are regarded as 2<sup>nd</sup> generation neural network models [33,34]. Because ANNs pursue practical approaches that borrow some key concepts from biological neural networks, such as parallel and distributed processing, ANNs do not strictly follow real biological functions. Generally, the key computing units of ANNs consist of neurons and synapses which mainly perform computing processing and store weight signals, respectively [35]. The functions of neurons mainly have been realized by using CMOS logic circuits for multiplication, addition, and activation. On the other hand, the functions of the synapses mainly have been established by using memory devices such as SRAM and e-DRAM for weight storage in neural chips [36-38]. However, conventional SRAM and e-DRAM have the weaknesses of volatility and large energy consumption [39].

Recently more bio-plausible spiking neural networks (SNNs) have been investigated and developed by mimicking the real functions of biological neural systems such as synaptic plasticity, spatio-temporal coding, integration and fire (I&F) neurons, and so on [40,41]. SNNs have emerged as a 3<sup>rd</sup> generation neural network model for future neural networks which are more energy-efficient than ANNs. For this approach, many types of silicon neuron and silicon synapses have been developed using conventional CMOS silicon circuits, but their size and energy consumption are much larger than biological neural systems [42].

As previously mentioned, conventional memory synapses and silicon circuit neurons are not satisfactory for both ANNs and SNNs, therefore more innovative memory device technologies have been investigated. Recently, memristor devices have received attention because of their outstanding characteristics such as non-volatility, scalability, small form factor, good reliability, low power operation, high performance, and their distinctive operation functionality, which is well-matched to

neural networks [43-46]. As expected, memristor-adopted neural networks show excellent performance and usefulness. Therefore, the use of memristor technologies for neural networks has been widely studied, from fundamental research to cost-effective manufacturing. In a short period, many research studies and neural network applications using memristors have been introduced, which has inspired new ideas but has also led to confusion. The primary causes of confusion arise from two areas. One is related to memristor devices themselves, and what their true technical status to support neural networks is, and the other is related to the relationship between memristor devices and neural networks for realizing desired functionalities. Already the possibility of many memristor devices as neural devices has been proven by estimating their key characteristics at the research level, which will be discussed later in detail [47-51]. However, their possible applications are different, and must be evaluated on a device by device basis, which is very closely related to the readiness and maturity of the devices that have been developed. For estimating the level of possibility to practical applications, the variability of a device can be a good reference if considered as an array in prototype level and not just one cell in research level, because device variability is the one of hurdles to overcome for transition of a device from being simply a possibility to the real product level. Also, we will discuss how the characteristics of memristor devices affect and relate to the functionality of different neural networks.

In this paper, we will focus on helping the reader to understand the current status of memristor devices and neural networks based on not only research related to possibility but also on maturity related to development for production. We briefly introduce memristor devices comparing the characteristics of prototype and research level, which will be helpful to compare memristor devices fairly based on maturity of devices in Section 2. The various types of neural networks such as currently available ANNs and more bio-plausible SNNs are briefly reviewed in Section 3. The advantages and challenges of the memristor devices adopted by ANNs are discussed in Section 4 and their interesting phenomena and usage for more bio-plausible approaches emulating biological functions are discussed in Section 5. Finally, we conclude this work with a perspective of memristor neural networks. The discussion will provide a perspective on the contributions and challenges of memristor-based neural network

technologies based on not only research and possibility, but also on development and practical applications.

## 2. Memristor Devices

In 2008, Strukov et.al. reported that the long-missing memristor was found [4] (but the claim was open to dispute owing to lack of a predictable relationship between electric charge and magnetic flux [52,53]), by using  $\text{TiO}_2/\text{TiO}_{2-x}$  based resistive random access memory (RRAM). Among emerging non-volatile memory devices, some devices such as magneto-resistance random access memory (MRAM), phase-change random access memory (PRAM), resistive random access memory (RRAM), and ferro-electric random access memory (FeRAM) emerged as memristor devices later [54-60]. As mentioned in the Introduction, they were initially developed as non-volatile memory devices before the first observation of a memristor device in 2008.

When we consider the maturity, scalability, and performance of memristor devices, MRAM, PRAM, and RRAM are representatives of memristor devices currently. Therefore, the possibility of other types of memristors has not been proven and is under investigation, so we mainly focus on MRAM, PRAM, and RRAM as memristor devices in this paper. For example, even though the FeRAM has already been developed and produced as a memory device in some companies (e.g., Fujitsu, Ramtron)[61,62], the memristor operation in the FeRAM is still under investigation [63]. As shown in Fig. 1, MRAM, PRAM, and RRAM have explicit pinched-hysteresis loops in their I-V curve and these are regarded as characteristics of memristor devices. The characteristics of these three devices stand out owing to their totally different operation mechanisms [64-66].

Therefore, when applying the characteristics of each device to appropriate applications, it is not easy to compare performance fairly by considering only the device characteristics at the research level without considering the maturity of the technology. Some memristor devices have been commercialized, but many remain under investigation as candidates for commercialization. Therefore, there is a still heated debate when comparing the superiority of specific device characteristics fairly [67,

68]. To overcome this confusion, we need to establish a standard methodology for comparison. Usually the characteristics reported in research level are much better than the characteristics in prototype chips because characteristics in research papers mainly reflect a device's future possibility, and related data are collected from single cell or small arrays, which do not sufficiently consider commercial production aspects. This tendency sometimes makes people overestimate the performance of new emerging devices based on data at the single cell level. To date, various emerging devices whose characteristics are excellent at the research level have been introduced, but many have never reached the prototype level. Only a few have been evaluated to verify their performance as prototypes, which is the preparation and early commercialization stage before going to mass production. MRAM, PRAM, and RRAM have been proven at the prototype level, although many other different memristor devices are under research and may prove useful for development [69-74]. Therefore, we will focus on these three memristors as devices for neural network applications. To provide a fair comparison by considering variability and maturity of devices for neural network devices, we will introduce a new type of comparison table representing the characteristics of not only the research level, but also the prototype level, as shown in Tab.1. This methodology can provide insight by distinguishing the possibility (research level) and the maturity (prototype level) of the devices.

The characteristics of NAND flash are listed as a representative example of conventional nonvolatile memory devices in Tab.1. NAND flash memory has taken center stage for data storage applications owing to its high density and non-volatility, which supports high density storage ( $> 100$  Gb/chip) for the system [75]. Even though NAND flash is an appropriate solution for data storage applications, it is not considered as a suitable device for neural network applications, because NAND flash has weaknesses in that it cannot provide byte addressable operation, fast latency time, and compatibility with logic operation because of its high voltage operation ( $>20$  V), as shown in Tab.1. Thus, the current main device for neural networks has been implemented using SRAM instead of NAND flash, even though SRAM has significant weaknesses, such as large cell size and volatility [76]. On the other hand, all memristor devices introduced in Tab. 1 have much better performance features

required in neural network applications than the NAND flash device. Therefore, recently MRAM, PRAM, and RRAM have received attention, as they overcome the limitations of NAND flash and SRAM for neural network applications.

The characteristics of MRAM, PRAM, and RRAM at the prototype level are compared with NAND flash memory, the currently available conventional nonvolatile memory technology. As shown in Tab.1, MRAM, PRAM, and RRAM provide byte addressable operations, high speed latency, low power consumption, and good compatibility with logic operation, characteristics that make them good candidates as neural network devices. However, MRAM, PRAM, and RRAM differ in certain ways. Hence, it is very important to understand these to utilize them as neural devices.

The MRAM mentioned in this paper is the spin transfer torque MRAM (STT-MRAM) which is a type of MRAM based on the most advanced currently available technology to achieve higher scalability through smaller cell size [77,78]. As shown in the lower part of Fig.1, the magnetic tunnel junction (MTJ) which has a sandwich structure similar to an insulator inserted between ferromagnetic materials, is the key element for operating an MRAM. We can write and read the stored data in the MTJ, which has different tunneling probability, i.e., different resistance according to the magnetization direction of parallel or anti-parallel forces between the ferromagnetic materials of the MTJ. MRAM devices have achieved three major breakthroughs to achieve the current MRAM level. First, the insulator material of MTJ was changed from AlO to MgO, which led to a drastic increase of tunneling magneto-resistance (TMR) ratio of MRAM of over 200% at room temperature, and over 1000% at 5K by using a CoFeB/MgO/CoFeB MTJ structure [79]. Second, the writing method was changed from spin reversal to spin transfer torque (STT), which reduced the writing energy and simplified the cell structure drastically, as shown in Fig. 2(a). The cell switching current increases drastically as cell size decreases in conventional MRAM, which is a major problem for scaling cell size. After changing the writing scheme from spin reversal to STT, the increase of switching current as cell size decreases essentially disappeared. Third, ferromagnetic materials were changed from planar MTJ to perpendicular MTJ (p-MTJ) having PMA (Perpendicular Magnetic Anisotropy). Transition materials such as Co and Fe, when



formed as multilayers in combination with noble metals such as Pt, Pd, and Au, show large PMA [78], which enhances the thermal stability and cell scalability of STT-MRAM. This type of PMA is categorized as bulk PMA in contrast to interfacial PMA. In 2010, Ikeda et al. found the interfacial PMA (i-PMA) in CoFeB/MgO-based MTJs [80]. It is very meaningful for real applications that they demonstrated a large TMR and a large PMA with a relatively low STT switching current, even though PMA has been observed at transition metal/oxide interfaces before [81]. As we discussed previously, an i-PMA MTJ structure using CoFeB/MgO is very similar to the conventional MTJ structure, which allows researchers to utilize their knowledge and experience already established when developing previous MRAM technology. Therefore, current R&D for manufacturing has focused on i-PMA MTJ owing to its compatibility and extendibility with conventional MRAM technology. By developing a novel damage-less MTJ patterning process, robust magnetic and electrical performances of i-PMA MTJ cells down to 15 nm nodes can be achieved [82,83]. Therefore STT-MRAM devices have achieved 256 Mb density at the prototype level and have been evaluated to achieve 4 Gb at the research level [77]. From now on, we will use the term of MRAM instead of STT-MRAM for convenience in this paper. In the case of prototype level MRAMs, write performance is very fast ( $< 50$  ns) and endurance is very high ( $> 10^{12}$  cycles) [84,85], which are almost the same as SRAM performance. Also MRAM shows the possibility of extremely low voltage operation (0.27~2.4V) [86,87] and low energy consumption ( $6.24 \times 10^3$  fJ/bit), as shown in Tab.1 [88]. However, MRAM density is very low compared to PRAM. Ideally MRAM can achieve a  $6F^2$  cell which represents one transistor as a selector and one MTJ as a memory element, but currently real cell size in an array is much larger than that in ideal case. Recently, the most advanced MRAM technology having a feature size of  $46F^2$  at a 28-nm technology node was reported [69]. This large cell size of MRAM results from fabrication difficulties at the array level. The MTJ, which is the key storage element of MRAM, consists of a very complicated structure required to achieve higher TMR ratio and better distribution. This structure contains a very complicated multi-layer of metal oxide and metal layers including refractory materials such as Ru, CoPt, and so on. It is very difficult to etch these materials, which leads to a much larger area than the ideal cell size [89].

The main advantage of PRAM is that we can achieve relatively high-density due to a cross point cell having small foot print ( $4F^2$ ) and the capability of cell stacking integration, so a 128 Gb density PRAM device has already been introduced in the market for memory applications [71]. However, PRAM has the weakness of relatively slow write speed ( $\sim 500$  ns), large energy consumption ( $\sim 30$  pJ), and limited endurance ( $10^9$  cycles) currently, although we can achieve much better performance at the research level, as shown in Tab.1 [71,72].

The key factor in PRAM to achieve energy efficient operation for neural network applications is to achieve fast write speed without any degradation of any other performance parameter. The current phase change material system for commercial product is mainly based on GeSbTe system, which has been developed and is a well-known basic material for PRAM [90]. GeSbTe is a ternary compound with composition  $\text{GeTe-Sb}_2\text{Te}_3$ . In the GeSbTe system, there is a pseudo-line along which most of the alloys lie. Moving up and down this pseudo-line, it can be seen that as we go from  $\text{Sb}_2\text{Te}_3$  to GeTe, the melting point and glass transition temperature of the materials increase, write speed decreases, and data retention increases. Because of the trade-off limitation between retention and write speed modifications of GeSbTe and totally different material system as PRAM material group have been attempted, as shown in Fig. 2(b). The modified GST through use of materials such as N and C improves PRAM retention and speed over the original GST [91,92]. To achieve more speed using less energy, totally new structures such as a  $\text{GeTe/Sb}_2\text{Te}_3$  super-lattice (SL) has been proposed. The operation mechanism of SL does not require the melting process that consumes large energy and slow crystallization time involved in conventional PRAM operation [93,94]. A 10 ns write speed and 10 years retention time at  $70^\circ\text{C}$  are achieved with the SL structure. Also, new materials such as GeBiTe [95] and ScSbTe [96] have been investigated to improve the write speed of PRAM as shown in Fig.2(b). Surprisingly, the write speeds (actually crystallization time) are less than 30ns. Especially, ScSbTe shows sub- nanosecond crystallization time with 10 years retention time at very high temperature. Rao et.al. revealed that “scandium is the key” for high performance of ScSbTe by using density functional theory (DFT) calculation [96]. Adding the element creates strong bonds with neighboring antimony and tellurium

atoms, forming cube-shaped nuclei that remain intact even when enough electricity passes through the material to raise its temperature to 600 K, which promotes a fast switch between the amorphous and crystalline phases. Also people achieve very good possibility of low voltage operation ( $<1.5\text{V}$ ) [97] and extreme scalability of cell size ( $\sim 2.3\text{nm}^2$ ) [98] for future PRAM. However, these results were acquired at the research level, where we can only evaluate the possibility of PRAM and additional time and resources will likely be required to reach product level technology.

Fundamentally, the characteristics of RRAM are very similar to PRAM. Comparing the performance values at the research level of PRAM and RRAM, they exhibit small differences, as shown in Tab.1. RRAM devices have a very simple structure, where a metal oxide is sandwiched between two metal electrodes. Since their first appearance, many RRAM devices have been studied and various materials and structures have been investigated owing to their simple integration, low temperature process, and scalability for high density devices [101-103], although it is very difficult to categorize and analyze them simply. Regarding their operation mechanism, RRAM devices can be classified roughly into two types: oxide-RAM (OxRAM) and conductive bridge RAM (CBRAM). The difference is that OxRAM's filament consists of oxygen vacancies in the oxide layer ( $\text{HfOx}$ ,  $\text{TaOx}$ ,  $\text{TiOx}$ , or  $\text{AlOx}$ ), while CBRAM's filament consists of metal atoms formed by fast-diffusive Ag or Cu ions migrating into the solid-electrolyte, as shown in Fig. 2(c).

Usually OxRAM devices have a low on/off resistance ratio (in the range of 10–100) and offer good endurance (up to  $10^{12}$  cycles), while CBRAM's on/off resistance ratio can be quite large ( $10^3$ – $10^6$ ) but has limited endurance ( $<10^4$  cycles) [111-112]. The key challenge of RRAM cell design is the variability of the switching parameters; owing to the stochastic nature of ionic (oxygen vacancies or metal ions) migration, the filament shape varies from device to device, and from cycle to cycle (within one device). RRAMs have issues with reproducibility of their electrical characteristics; there are large resistance variations not just between devices, but also between cycles of programming on the same device [113-115]. This problem has been holding RRAM back from commercialization despite its many attractive features.

Although the density of RRAM prototypes for the commercialization stage is very low (~4Mb) from its large variability, recently researchers have made good progress and shown the possibility of high density devices that achieve outstanding characteristics such as low power consumption(~0.1pJ)[110], better reliability(endurance  $>10^{12}$ , retention  $> 10\text{yr}@150\text{C}^\circ$ )[108-109], and higher density(32Gb) using a simple fabrication process, and strong thermal properties [101-102].

Moreover, the main methodology of neural computing based on deep learning is stochastic computing such as regression, and pattern and speech recognition, which requires less-precise computing than deterministic conventional computing. Thus, the variation of memristor devices for neural applications has less effect on the result of computing compared to memory application, which allows memristor devices, especially RRAM, more opportunities for neural applications [116]. As we discussed, although RRAM devices have some issues related to variability, the more tolerable requirement for variation in neural applications and recent technical progress make RRAM devices good candidates for neural devices.

Until now we have discussed the current technology status of memristor devices for neural applications and mainly consider three memristor devices, MRAM, PRAM, and RRAM for neural network applications, which are very close to mass production in the semiconductor industry and have desirable core competencies over conventional NAND flash. After discussing neural network technologies in the next section, we will further discuss how to utilize these memristor devices in neural network technologies and applications based on real technical properties of memristor devices including variations of real devices, not just ideal expectations.

### 3. Neural Networks

Current computing architecture was proposed by Von Neumann in 1945 [117] and developed drastically, keeping pace with the advances in semiconductor technology represented by Moore's law [118]. However, the advent of the big data era involves processing tremendous amounts of data, whose uncertainty also increases rapidly [119]. This trend leads to strong motivation to change the paradigm

of computation from simply increasing speed to increasing efficiency. In the early computing era, besides the Von Neumann computing architecture, there were attempts at different type of computing including thinking machines, such as Turing machine [120], followed by studying the efficient computing technology inspired by the human brain, which is known as AI, which was used for the first time as the topic of the second Dartmouth Conference, organized by John McCarthy and colleagues in 1956 [121]. It is known that the efficiency of the human brain is essentially 500,000 times better than that of a conventional computer [122]. Therefore, people have taken note and have tried to mimic brain functions for efficient computation, which is the main motivation behind AI technology. Even though historically there have been up and down periods in research and industry, the technologies and applications of AI. have been developed continuously and have achieved considerable progress. Eventually Alpha-Go defeated a human being at the highly popular game of Go, which is partly responsible for the surge in interest in AI [123,124].

Nowadays many technologies have been introduced explosively in the AI field, many of which are not mature technologies, so there is some confusion in explaining AI, machine learning, and deep learning. Even though there are still many different opinions to explain the relationship among these topics, generally the relationship among them can be regarded that AI is a broad concept encompassing all intelligent computing, machine learning is a type of methodology involving computing by learning from data rather than by programming, and deep learning is a type of specific technology used to train computing systems to be smarter by mainly using deep neural network technologies, as shown in Fig. 3 [125]. The idea behind AI is simple and broad, yet fascinating, which is to create intelligent machines that can make decisions on their own. Machine learning is a subset of AI that provides computers with the ability to learn without being explicitly programmed by exposing them to vast amounts of data. The core principle behind machine learning is to learn from data sets and try to minimize error or maximize the likelihood of their predictions being true. Deep learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain, called artificial neural networks. Deep learning automatically extracts the features which are important

for classification from data, whereas in the case of machine learning, these features must be manually defined. Nowadays neural networks have been highlighted as technologies to be applicable for not only deep learning problems but also many wider AI applications [126-128].

Since the introduction of the perceptron by Rosenblatt in 1958 [129], neural networks have made progress by overcoming the self-contradiction problem and other limitations [130-131]. In 2006, Hinton, et. al. introduced a multi-layered feedforward neural network that could be effectively pre-trained one layer at a time, treating each layer in turn as an unsupervised restricted Boltzmann machine, then fine-tuning it using supervised backpropagation [132]. After this substantial progress, deep neural networks have received increased attention and have provided solutions for many AI applications such as pattern recognition, natural language processing, forecasting and prediction, speech recognition, bio-informatics, autonomous vehicles, robotics, and so on [133-152].

Numerous ANNs have been developed by modifying and adjusting architectures for more accuracy and efficiency according to different applications [153], but backbone architectures among neural networks are multi-layer perceptions (MLPs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs), as shown in Fig. 4 [285]. MLPs are basic feed forward deep neural networks, which consist of an input layer, one or more hidden layers, and an output layer. MLPs can be trained by back propagation (BP). Main usage of MLPs having 3–10 neuron layers involves simple classification, such as hand-written pattern recognition [154-156]. CNNs are designed to process data that come in the form of multiple arrays, for example a color image composed of three 2D images containing pixel intensities in three color channels. CNNs take advantage of properties of signals: local connections, shared weights, pooling, and the use of many layers. CNN has achieved many practical successes and has been widely adopted by the computer vision community [157-161]. CNNs are usually very deep neural networks having 5–100 layers, and their main applications are in vision processing such as face recognition and image classification. RNNs have been established as a network of neurons with feedback connections. Because RNNs include feedback loops, they can store information while

processing new inputs. This memory makes them ideal for processing tasks where prior inputs must be considered (such as time-series data) when dealing with the current inputs. Therefore, they can learn many behaviors including sequence processing tasks, and algorithms and programs that are not learnable by traditional neural network learning methods. The main uses of RNNs are sequential data processing such as speech recognition, translation, and natural language processing [162-166].

Drastic advances in neural network technologies leads to innovative computing, which is energy-efficient and human brain inspired. This leads to changes in computing architecture and devices, not only software. New types of devices having much better characteristics than conventional memory devices such as SRAM and DRAM have been investigated, and recently the memristor devices mentioned in previous sections, such as MRAM, PRAM and RRAM have been considered as neuromorphic devices for neural networks. It has been found that memristor devices also have good characteristics to support not only ANNs but also SNNs [167-172]. We will discuss how to utilize memristor characteristics for both of ANNs and SNNs in the following sections.

#### 4. Memristor Devices for Artificial Neural Networks

The advance of ANNs makes computing systems smarter, which provokes changes in computing architectures, involving not only software but also hardware, to implement neural networks in systems. People have utilized various hardware types to execute neural networks, from conventional computers, neural accelerators, to neuromorphic computers [173-185]. In fact, neuromorphic computers are the most efficient method to execute neural networks, which require different types of computing devices, so called neuromorphic chips or neuromorphic processing units (NPU). The main difference between central processing units (CPUs), graphics processing units (GPUs), and neuromorphic processing units (NPUs) is the number and the organization of computing cores, as shown in Fig. 5(a). Comparing conventional CPUs which have at most several tens of cores, NPUs and GPUs have many cores, over several thousand [186]. NPUs and GPUs are very similar in that they use many cores for parallel processing in neural networks and image processing applications respectively, but there is a significant

difference in operation and organization. The cores of GPUs work mainly as processing elements (PE) and do not have memory function. Although GPUs have SRAMs, their main function is a cache. GPUs need external memory function, which means processing and memory operate separately in GPUs, like in CPUs. On the other hand, the cores of NPUs have their own memory units, so called synapses, to execute the neural network operation, which is different from the core architecture of CPUs and GPUs. Therefore, NPUs operate processing and memory simultaneously during computation, which is very similar to computation in the human brain. But currently many NPUs use volatile devices such as SRAM or DRAM as synapses [187,190,191]. Even though their processing and memory operation is associated, they require external storage like Flash memory or HDD having non-volatile properties to store weights. If we use memristor devices as synapses which have non-volatility and do not require external storage, we can realize simultaneous processing and memory in computation [193].

NPU architectures pursue the associated functions of processing and memory, which reduce energy consumption and data congestion drastically during data transfer between processing elements and memory devices. Fig. 5(b) shows a comparison of the computing energy efficiency among CPUs, GPUs, and NPUs. The energy efficiency increases in the order of CPUs, GPUs, and NPUs. NPU data displayed in this figure are collected from the performance of the deep neural network processors published at the International Solid-State Circuits Conference in 2016 and 2017 [188]. Overall, efficiency is a key for neural hardware: NPUs must be built with dramatically improved efficiency in size, computation, data movement, and energy compared to conventional computing chips.

#### 4.1 Memristor synaptic devices for weight storage

An ideal neural chip must be equipped with highly parallel processing capability for bit-wise, fixed- and floating-point computation of various data lengths, a memory bandwidth orders of magnitude greater than what is available in today's conventional computing, extraordinary amount of memory able to handle big data processing, low memory latency, and novel architecture allowing flexible and rich



connectivity between computing elements and memory. All of the aforementioned characteristics must be implemented with very low power consumption and high energy efficiency.

Fig. 6 shows how the idea of a neural network unit can be implemented in circuit design within NPU cores. The basic unit to constitute neural networks is shown in Fig. 6(a). Input neurons (neurons in the input layer) having an input value  $X_i$  coming from the presynaptic neurons will conduct matrix multiplication with synaptic weight  $W_{ji}$  and generate input values,  $Z_j$ , to the hidden neurons or output neurons. Eventually the output values of hidden neurons or output neurons,  $Y_j$ , will be generated it will work as input values of next neuron layer. This neural network scheme can be realized through semiconductor technologies as shown in Fig. 6(b). Generally, the function of input/output layers (neurons) including multiplication, addition, and activation can be established by CMOS digital or analog logic circuits, and a weight storage function (synapses) can be established by memory devices such as SRAM or DRAM, which store the updated weight values. In this approach, the key role of conventional synaptic devices (SRAM, DRAM) is storing weights. Thus, this approach for synaptic function is called weight storage of synaptic devices [189].

Recently many neuromorphic processors including accelerators have been developed, as shown in Tab. 2. Even though their approaches are a little different in technology node and neuron device types, their synapses are generally achieved by volatile memory devices such as SRAM and DRAM to store weights. SRAM has a severe weakness for practical synapse device because it has very large cell size which prevents NPUs from achieving high density synapses. Furthermore, SRAM and DRAM are volatile, which makes the operation of NPUs complicated and inefficient owing to the loss of their trained weights after system power-off [36,38,190,191]. Nevertheless, they are necessary because until now, only these devices have been available for neural network applications. However, memristor devices have been receiving attention as alternatives because of the remarkable progress in development. As we previously mentioned, MRAM and PRAM among many memristor candidates are now ready for practical neural network applications. Even though RRAM has strong potential, it presently requires more work for practical applications, and it will be discussed for more advanced

applications as future work, such as matrix-vector multiplication and SNNs. In this section as a practical approach to weight storage, we will mainly consider MRAM and PRAM as synapses. We describe the characteristics of devices related to neural network functions in Tab. 3 by referring to conventional SRAM synapses [38,69,71]. The essential characteristics of memristor devices as synapses are good process compatibility with standard logic technology and non-volatility. The architecture of neural networks has many more interconnections than conventional computing, for example a simple fully connected neural network for the MNIST dataset, consisting of one input layer, one hidden layer, and one output layer requires 203,264 ( $784 \times 256 + 256 \times 10$ ) interconnections, compared to the 32 or 64 bus lines used in current computing architectures. Energy loss of passing through interconnections is estimated as  $E = 1/2 \cdot C_L \cdot V^2$ , where  $C_L$  and  $V$  denote parasitic capacitance of interconnection lines and applied voltage. Interconnections in the system can be categorized as three parts: on-core (inside core), on-chip (interconnections between cores inside chip), and off-chip (interconnections between chips). Length of interconnection lines from on-core through on-chip to off-chip increase drastically, and naturally capacitances increase in same manner. The increase of capacitance is almost 10 times higher between each step from the on-core through the on-chip to the off-chip. Also applied voltages increase from on-core to off-core and to the off-chip several times. Eventually energy consumption according to interconnection categories increases by a factor of 100 from the on-core to the on-chip operation and additionally by a factor of 100 from the on-chip to the off-chip operation [192,193]. Therefore, compatibility with CMOS logic technology is an inevitable requirement of memristor devices for neural networks to minimize the energy consumption of massively parallelized and distributed network interconnections.

Also, non-volatility of memristor devices as synaptic devices is one of the essential requirements for neural networks. After training the neural networks, all results of trained weights should be stored. However, if synaptic devices are not nonvolatile (SRAM or DRAM), they lose the trained weights after system power off. Thus, they need separate storage devices having non-volatility, such as NAND Flash external to the neural chips. As previously mentioned, these types of neural systems consume large

amounts of energy and wake time during power on because numerous weights must be uploaded to neural chips [193]. To make matters worse, this operation requires a programming strategy to escape problems such as sudden power off recovery (SPOR) during accidental power-off, which makes the system complicated and inefficient [194,195,289].

MRAM and PRAM have good process compatibility with standard logic process and are non-volatile, so they meet the basic requirements as synaptic devices. Currently, advanced technologies for neuromorphic chips, such as IBM's True-North, use 256 Mb SRAM cells as synapses based on 28 nm technology nodes. As shown in Tab. 3, SRAM size of these technologies is estimated between 0.15–0.2  $\mu\text{m}^2$ . The prototypes of MRAM and PRAM applicable to synaptic devices have very high density and nonvolatile characteristics, previously mentioned in Tab.1 [69,71]. If we apply these technologies, we can achieve 376–500 times and 4–5.5 times higher density of synapses by using PRAM and MRAM than SRAM synapses, respectively, with the advantage of nonvolatile characteristics. However, to find proper usage of MRAM and PRAM, we need to evaluate the memristor devices based on performance, not only on density. As shown in Tab. 3, write latency and endurance of MRAM are almost the same as those of SRAM. But PRAM exhibits weaknesses in some performances. The endurance is limited to  $10^9$  which is not sufficient to fulfill the requirements of on-line learning, and write latency ( $\sim 500$  ns) is much slower than that of SRAM [64]. Although the performance of PRAM is not suitable for some high-performance applications, it has a very strong advantage in high density applications for very deep CNNs. For instance, VGGNet, consisting of 19 layers, has 24 million neurons and 144 million synapses and requires over 1.2 GB memory size for 32-bit floating point operations using batch size 1 [196]. Therefore, it is very difficult to implement large scale CNNs in mobile phones unless PRAM is the synaptic device because of memory requirements. When we use PRAM as a synaptic device, slow latency time and large energy during write operation cause degradation of performance and energy efficiency. To overcome these issues, the overall write performance of PRAM has to be improved. As we discussed in chap.2, many studies about phase change materials show big progress and the possibility, but it takes time for new materials to become prototype level quality [91-96].

As an alternative approach for PRAM to be practically available as a weight storage, we can try to find the methodologies to maximally utilize the advantages of PRAM, such as high density with minimizing its weakness. The neural hardware system can be divided into inference and training units. Considering the computing procedures of ANNs, write operations mainly occur in training. During inference, mainly read operations take place [223-225]. The main drawbacks of PRAM come from write operations. Despite this limitation, using PRAM synaptic devices can be the best solution for high density inference machines to perform feedforward inference efficiently where read operation mainly occurs. For example, this can be appropriate for mobile phone applications [197]. Related applications will be discussed more in 4.2 section.

Thus, we can utilize PRAM as a high density synaptic device(weight storage), especially for inference machines, and MRAM as a high performance synaptic device(weight storage) in neural networks with more synapses and higher energy efficiency (including non-volatility) compared with SRAM synapses.

## 4.2 Memristor devices for matrix-vector multiplication

Until now we have discussed the application of memristor devices for weight storage, which is currently the most popular methodology for practical neural processing units (NPU). Now researchers are investigating methods to utilize memristors more effectively and innovatively by taking advantage of unique characteristics of memristor devices, i.e., nonlinear I-V characteristics.

A particular operation, matrix-vector multiplication, where matrix elements having analog values or multi-level cell (MLC) values has been attempted to achieve better efficiency in ANNs by using more functionality of synaptic devices beyond the role of simple weight storage in neural networks [198]. The memristor crossbar array can naturally transfer the weighted combination of input signals to output voltages and accomplish matrix-vector multiplication efficiently by reducing the computation complexity from  $O(n^2)$  to  $O(1)$ . As shown in Fig. 7, the relationship between the input voltage vector ( $\sim V_i$ ) and output voltage vector ( $\sim V_o$ ) of a memristor (RRAM for this figure) can be expressed as

follows [198]:

$$V_{oj} = \sum w_{jk} * V_{ik}, \quad w_{jk} \rightarrow 1/M_{jk} (\approx G_{jk})$$

where  $k$  ( $k = 1, 2, \dots, N$ ) and  $j$  ( $j = 1, 2, \dots, M$ ) are the index numbers of input and output ports, and the matrix parameter can be represented by the conductance ( $G_{jk}$ ) or inverse of the resistance ( $M_{jk}$ ) of the memristor devices. The continuous variable resistance states of memristor devices enable a wide range of weight matrices that can be represented by the crossbar shown in Fig. 7(a). During this matrix-vector multiplication procedure, memristor devices can naturally execute multiplication and addition simultaneously without any additional logic devices such as a multiplier and adder owing to their unique memristive characteristics. This unique computing characteristic is called matrix-vector multiplication operation, where one memristor plays a composite role of one weight storage, one multiplier, and one adder [199]. To support this operation appropriately, it is required that the memristors have good proportionality between conductance ( $G$ ) and voltage ( $V$ ) [200-202]. Fig. 7(b) shows the conductance values as a function of set voltage following SET-RESET switching in a TaOx based memristor device. This shows good proportionality, which can be modeled by a quadratic polynomial. Fig. 7(c) shows a representation of the procedure for programming the memristor device. For clarity, considering only two conductance targets ( $G_1$  and  $G_2$ ), we can control the conductance of a memristor by applying various voltages through TE, GE, and BE. To obtain the target value of  $G_1$ , the conductance state was moved from SET to RESET by gradually changing the voltage of GE and BE nodes and the  $G_2$  state was achieved from RESET to SET in Fig. 7(c).

We can execute matrix-vector multiplication and train neural networks using this methodology [203,204]. Currently memristor devices are very appropriate to support this variable conductance technology, which enables neural networks to be operated as matrix-vector multiplication in neural chip.

A comparison of circuit and simulation results for different neural networks based on weight storage and matrix-vector multiplication operation technology is shown in Fig. 8. As shown in Fig. 8(a) and 8(b), the architecture of memristors for matrix-vector multiplication operation is much simpler than for the weight storage scheme. It is obvious that computing by using NPU is much more efficient than by

using conventional processing units; moreover, this more aggressive approach (matrix-vector multiplication) leads to extremely efficient computing. For some applications such as certain examples of DNN and edge detection, the 1/24–1/28 area reduction and 196–308 times energy efficiency can be achieved by using matrix-vector multiplication in comparison with SRAM based weight storage [205]. If memristors are only used for weight storage, SRAM can be replaced by memristor devices without having to change current architecture. Therefore, we can take advantage of the intermediate area reduction and energy efficiency located between matrix-vector multiplication and weight storage of memristors based on current SRAM functions (refer to tables in Fig. 8). The matrix-vector multiplication approach is the most efficient way to utilize memristors as synapses for neural networks, but there remain problems to be solved for real applications. Matrix-vector multiplication utilizes analog characteristics or sufficient multi-levels of memristor devices to support neural networks for accuracy.

It has been shown that the inference procedure called forward propagation in neural networks does not require high resolution of weights relatively, so recently binary neural networks (BNNs) have been investigated as a cost-effective and energy efficient solution for inference machines [206-210]. On the other hand, for online training when weights are updated on the hardware during run-time, the training procedure in neural networks requires much higher resolution of weights than the inference procedure. The reason is that neural networks perform derivatives for gradient descent during back-propagation, which requires continuous values. Back-propagation also passes even small training errors from the output layer to the input layer, so if the precision is insufficient, such small errors will be skipped, deteriorating the accuracy of the network. Fig. 9 shows the accuracy vs. resolution of weights/neurons for different cases. At least 6 bit-resolution is required for MNIST classification in MLP [210], but the requirement for MNIST classification in CNN increases up to 10 bits. More complicated applications, such as traffic sign classification (GTSRB database,) require much higher resolution of weights [211]. Even though resolution depends on the type of application, the memristor based matrix-vector multiplication operation requires at least 6 bits (64 levels) resolution, i.e. a 64 multi-level cell (MLC) in a memristor cell, to achieve useful accuracy [210-212].

The possibility of MLC operation in memory devices depends on the on/off ratio of the sensing signal, variability of devices, density of operation, and so on. Currently the memory device having the highest multi-level operation is the NAND flash device, of which available multi bits are at most 3 bits (8 levels). Moreover, MLC operation in NAND flash device requires special circuit design and software which further requires tracking time and energy to set parameters precisely [213].

It is very difficult for current memristor devices to achieve sufficient multi-level operation at a 64 level MLC, so we need to find the optimized operation region for accuracy between the device and neural network. If we increase the multi-level of memristor cells to achieve better resolution of neural networks the accuracy of memristor devices will decrease owing to the increase in distribution and uncertainty and vice versa. Therefore, practical neural networks encompassing matrix-vector multiplication with memristor synapses can be associated to learning, where adoption is based on a functional equivalence existing between deterministic learning rules of neural networks with a multi-level memristor, and probabilistic learning rules with a binary memristor [214].

Even though many memristors show the possibility of MLC operation at the single cell level [215,216], a sufficiently high order ( $> 6$  bits) of MLC memristor devices have not been developed yet at the prototype level. This means that current memristor technologies for matrix-vector multiplication operations cannot provide sufficient multi-level cells for neural networks using deterministic learning rules. Therefore, practically we should consider the neural networks with probabilistic learning rules such as binary and ternary neural networks for matrix-vector multiplication using memristor devices until successful development of higher order ( $> 6$  bits) MLC memristors [217].

For this reason, people have been interested in BNNs which have binary weights and activations. Some researchers are trying to generalize this concept to quantized neural networks (QNNs) having low precision weights and activations such as binary, ternary, and slightly higher [218]. In this review, we will use the meaning of BNNs as representative of neural networks having low resolution, including QNNs, for convenience. It is expected that BNNs substantially improve energy-efficiency and availability with memristor devices, because most arithmetic operations in BNNs can be replaced with

single-bit operations [219]. On the other hand, the resolution of BNNs is very low, which makes the effectiveness of BNNs questionable. Therefore, many researchers have tried to validate the effectiveness of BNNs and have achieved nearly state-of-the-art results over datasets in some image classification application from the datasets of MNIST, CIFAR-10 and SVHN [220-222].

Even though BNNs cannot cover all applications owing to the limited resolution, in some special applications which can be accomplished with the BNNs, memristor based matrix-vector multiplication technology can be realized in the near future. In fact, neural hardware such as chip and system can be divided into inference and training units, as shown in Fig. 10(a). Especially in inference units, BNNs with memristor based matrix-vector multiplication technology are very effective and are practically achievable with very low technical barriers. Therefore, some companies such as Google and NVIDIA have developed some neural hardware consisting of separate inference and training units [223-225]. If clients using mobile phones, tablets, and PCs are equipped with neural devices that have only the function of inference, they can form neural networks with BNNs which lead to drastically reduced memory size and computing energy [226]. Furthermore, clients can update the binarized training results from servers or data centers, as shown in Fig. 10(b). If we apply memristor based matrix-vector multiplication technology in inference units with BNNs for mobile phones, we can supply the best solution for extremely low energy and cost-effective operation.

The hardware supported by separate inference and training units is operated based on off-line learning, not on-line learning. The matrix-vector multiplication operation in on-line learning requires over 6 bits MLC operation of memristor devices, which is too challenging to be realized in current memristor technologies [210].

In this chapter, we have focused on real application of ANNs based on technically proven memristors such as PRAM, MRAM, and RRAM. NPUs in the development stage use SRAM and e-DRAM for weight storage with essentially the conventional role of memory devices. In memristors based weight storage, if we replace SRAM and e-DRAM with PRAM, RRAM or MRAM, we can acquire the advantages of high density, low power consumption, and non-volatility which are the



essential characteristics for neural networks [210]. Subdividing the main roles of memristors, PRAM is suitable for high density applications (very deep neural networks containing vast data) and MRAM is suitable for high performance and high endurance applications (very frequent change of data as in on-line learning). The matrix-vector multiplication operation using memristor devices has been proposed to achieve higher efficiency. For certain applications, BNNs can be used to avoid the difficulty of MLC operation of memristors, and BNNs with matrix-vector multiplication memristors for inference purposes can be an extremely energy efficient computing solution for some special applications (off-line learning). In the near future, it is essentially assured that memristor devices will contribute to the advance of ANNs and their applications practically by replacing conventional memory devices, as discussed in this section.

## 5. Memristor Devices for Spiking Neural Networks

ANNs, previously mentioned, are not strictly based on real biological operations but on the biological concept for efficient computing, such as eliminating bottlenecks in bus lines in conventional computing and by applying simultaneous operation of processing and memory. A different approach to follow the more bio-plausible operation has been studied as named Spiking Neural Networks(SNNs). The comparison between ANNs and SNNs are depicted in Fig. 11. The unique operation of SNNs is that the output of their neurons is a time-space-encoded pulses and the timing domain information is expressed through the membrane potential value, i.e., the membrane potential records the historically received and issued pulse energy [228,229]. Therefore, multiple neurons can realize the expression ability of space-time two-dimensional space [230]. There are many simulation algorithms for dynamic behavior of neurons, which are generally expressed by differential dynamic equations, and have good bionic capabilities, but are not conducive to hardware implementation. Therefore, a method based on a simplified algorithm, such as Leaky Integrate and Fire (LIF) model, has received much attention, as shown in Fig. 11 [231,232]. The principle is that the pulses connected to all axons of this neuron are weighted and summed according to the synaptic strength to obtain the integrated potential of the neuron,

which is then added and updated with the previous membrane potential to obtain a new membrane potential. The pulse is issued if the membrane potential exceeds the set threshold, otherwise it is not issued as shown Fig. 11(a). It can be seen that the LIF algorithm has the feature of expressing the temporal and spatial information jointly. The spike timing dependent plasticity(STDP) has been widely used as algorithm to train SNNs [233-235]. However, it is difficult to train the SNNs by using classically emulated STDP algorithm which requires complicated hardware. In practical applications, simplified STDP for better efficiency and back propagation algorithms for higher accuracy are also used to train SNNs [236]. Integrated circuits capable of learning by STDP rules generally require complicated, and large synaptic blocks hosting multiple transistors and capacitors [236,237]. To enable small-area synapse, hence, high-density neural circuits, memristor devices have recently attracted a strong interest [238-241].

To support the STDP learning algorithm in SNNs, it is required that the memristor synapse devices have the linear proportionality in  $G$ - $V$ (Conductance-Voltage) relationship at analog operation manner[116,260]. SNNs can potentially offer better energy efficiency for inference because the neurons in the networks are sparsely activated and computations are event-driven. Latency affects the computing performance more than throughput in event-driven computing as shown in Fig. 11(b), which is a requirement for memristor devices for SNN [227, 243].

## 5.1 Memristors for the STDP Learning Algorithm

A commonly used SNN learning algorithm is spike-timing-dependent plasticity (STDP) [244-246]. The synapses contribute to the computation by changing their connection strength because of neuronal activity, which is known as synaptic plasticity. The concept of synaptic plasticity has been heavily influenced by Hebb's postulate stating that the connection strength between neurons is modified based on neural activities in pre-synaptic and post-synaptic cells [242]. STDP is a more effective training algorithm that has been validated in the biological brain. As a locally trained, non-backpropagation algorithm, however, it does not guarantee obtaining high-performance networks.

Recently it has been reported that STDP for SNNs can be easily emulated using memristor devices such as PRAM, RRAM, and MRAM, which has accelerated the research in SNNs, as shown in Fig. 12 [245-251]. Fig. 12(a) represents the circuit diagram of 2-PRAM cells and the emulated result of STDP with a two-channel pulse generator as precise as 10 ns. Accordingly, STDP behaviors can be observed when  $\Delta C$ , related to conductance, is plotted as a function of  $\Delta t$ , as shown in Fig. 12(a) [17]. This result makes use of a communication signal between the pre-synaptic neuron and post-synaptic neuron, and the timing of the spike is tracked by the neuron circuitry, which mimics a form of Hebbian learning.

Also, several resistive change materials have been adopted as synaptic devices. The advantages of RRAM devices offer good scalability, fast switching speed, and low energy consumption. Until now, many types of synaptic devices based on RRAM such as GdOx/Cu:MoOx [252], TiOx [253], WOx [254], FeOx/SiOx [255], HfOx [256], AlOx [257], TiOx/HfOx multi-layers [258] and PCMO [259] have been reported. As an example of RRAM synaptic devices, Fig. 12(b) shows that STDP has been implemented in HfOx synaptic devices using different pulse schemes [256], which program resistive synaptic devices gradually. Another approach to realize the STDP function is shown in Fig. 12(c), which shows an MRAM based synaptic device with access transistors for separate spike transmission and learning current paths; STDP characteristics are achieved successfully [248]. Even though the structure and function of this neural device are quite different from typical memory devices, it is shown that MRAM technologies also can be utilized as neural devices.

There are two types of conductance change memristor synaptic devices for STDP. One is conductance modification by the sequential number of an identical pulse and the other is by the increasing voltage of the pulse, as shown in Fig. 13(a) [247,260] and 13(b), respectively [59,261,262] which require the characteristics of analog memristor devices. As modifying the conductance of the synaptic devices is directly related to the learning rule, such as Hebb's Rule and STDP, the different types of conductance modification can affect the neural network and can be interpreted differently. The voltage-driven change is capable of facilitating the STDP by forming adequate presynaptic and

postsynaptic spikes. For example, the timing of the presynaptic and the postsynaptic spikes can be transformed to a specific voltage effectively as shown in Fig. 13(d)[262]. Thus, this type can implement the temporal coding scheme more precisely, although the system becomes very complicated due to the complex forms of the spikes. In contrast, the number-driven change is devised to simplify the learning rule. In this case, the STDP can be approximated as a rectangular form, so the synaptic weight is simply increased or decreased to satisfy some condition of the timing. As shown in Fig. 13(c)[247], the simplified STDP can be interpreted as a simple learning rule, i.e., Hebb's rule, with the capability of long term depression. The synaptic weight modified by the identical pulse can make the system simpler than the voltage-driven one. However, the shape of the conductance change (especially the nonlinearity and the asymmetry in the G-V relation) can significantly affect the performance of the neural network as pointed out by G. W. Burr et. al.[260]. In spite of this drawback, the number-driven synaptic weight shows decent performance and high practicality based on experimental demonstrations, while the voltage-driven one has not been widely examined yet owing to the complexity and the difficulty of realization at the array level.

## 5.2 The emulation of More Neural Functionalities by using Memristors

Besides the STDP function, innovative technologies using memristors to achieve more functionalities mimicking brain functions such as Short Term Memory (STM)/Long Term Memory (LTM) and integrate and fire (I&F) neurons have attracted attention. As shown in Fig. 14(a), according to general cognitivist learning theories, human memory comprises the two key memory components of STM and LTM, the incoming information which is collected by our senses, such as visual and auditory sensors (sometimes regarded as sensory memory), is processed through STM. The key role of STM is to process information, so data in STM will be decayed (forgotten) after processing. However, LTM works as information storage more-or-less permanent retention in comparison with volatile STM. In this point of view, Conductive Bridge RRAMs (CBRRAMs) are very well-matched devices for emulating the transition between STM and LTM. CBRRAMs are another type of RRAM which relies

on fast ion diffusion. Fast diffusive ions such as Ag<sup>+</sup> and Cu<sup>2+</sup> migrate into the insulating medium materials to form a conductive bridge [263-265].

CBRAMs share many of the attributes of metal oxide RRAM, and have been utilized as synaptic devices. Some groups successfully achieved not only the STDP function but also transition from STM to LTM depending on the stimulus pulse intervals, as shown in Fig. 14 [266,267]. If intervals ( $>20$  s) are large enough, the device conductance decays as shown in Fig. 14(b) which can be emulated as a STM function. If the intervals ( $<2$  s) are small enough, the device conductance maintains a higher level without decay, as shown in Fig. 14(c), which can emulate a LTM function. This phenomenon can appear in many types of conductive bridge devices such as the Ag/Ag<sub>2</sub>S/nanogap/Pt [267], Cu/Cu<sub>2</sub>S/nanogap/Pt [268], and Ag/GeS<sub>2</sub>/W synaptic devices [269]. Generally, conductive bridge synaptic devices can naturally mimic many features of the biological synapses through the very similar operation principle of the release of ions into the junctions between biological neurons. To ultimately achieve brain-like neural networks, memristor devices that imitate biological neurons by maintaining the equilibrium potential, the transient dynamics, and the neurotransmission process is highly desirable. Moreover, researchers are emulating biological neuron functions beyond mimicking synaptic functions by using memristor characteristics [270-272]. Recently the prerequisite for PRAM-based neuron realization is to effectively simplify the complex biological neuron mechanism to be readily applied to hardware. The PRAM-based neuron must be able to 'fire' after receiving a certain number of pulses that can influence an internal state that does not necessarily relate to the external conductance unless the neuron fires when exceeding a threshold. Tuma, et. al. have recently reported a PRAM based neuron (Fig. 15) to integrate postsynaptic inputs [273], whereby the evolution of neuronal membrane potential as encoded by phase configuration within the device was demonstrated. More importantly, the ability to present remarkable inter-neuronal and intra-neuronal randomness using the devised neuron was also verified by using the switching mechanism of PRAM, especially the crystallization process. For PRAM-based neurons, the detection of temporal correlations within many event-based data streams was demonstrated, and a complete PRAM neuromorphic circuit made up of PRAM-based neurons and

PRAM-based synapses has also been delivered [272, 273]. These powerful functionalities from synaptic devices to neuro devices for SNNs using memristor devices should encourage further research of SNNs. SNNs utilize neuron models that communicate by sequences of spikes. The SNNs are an emerging model which encodes and processes information with sparse time encoded neural signals in parallel [274-278]. In order to understand how memristor devices are used to support these functions of SNNs, we have summarized the related research that is underway. Table 4 lists various SNNs consisting of the memristor devices discussed above[194,247, 279-284]. Although we can see that synapses primarily are consisted of memristor devices and neurons are made up of CMOS, even neurons have begun to study the entire composition of memristor devices for more energy efficiency and smaller foot print, as discussed earlier. The function of neurons is mainly based on Integration & Fire(I&F) or Leaky Integration & Fire(LIF). Training is being implemented mainly in STDP or simplified STDP for more practical approach previously mentioned. Also other learning rules called SRDP has been considered to be responsible for SNNs. In this learning rule a high frequency of PRE and POST spikes leads to potentiation, while a low frequency leads to depression [181,194]. The back propagation training which is mainly adopted in ANNs, not listed in the table, is also considered to increase accuracy for SNNs [285].

As a bio-inspired architecture abstracted from actual neural systems, SNNs not only provide a promising solution to deal with cognitive tasks, such as object detection and speech recognition, but also inspire new computational paradigms beyond the von Neumann architecture and Boolean logic. These technologies can promote drastic advances in performance and efficiency of computing systems. However, an energy efficient hardware implementation and the difficulty of training the models remain as two important obstacles that limit the application of the SNNs. However, SNNs are not a mature technology, and their motivation, bio plausibility, and efficient computing are very clear and challenging. Also memristor devices should inspire technical advances in SNNs and become appropriate solutions for both SNN and ANN devices.

## 6. Conclusions

Memristor devices have become popular research subjects because of their outstanding characteristics for not only memory device applications but also neural network applications. Although synaptic devices currently in use for weight storage in neuromorphic chips are conventional SRAM and e-DRAM, they have disadvantages regarding volatility and large cell size. Memristor devices such as PRAM, MRAM, and RRAM have the advantage of non-volatility and much smaller cell size (= higher density), characteristics that allow computing system to have higher density and to be more energy-efficient. Utilizing their distinctive characteristics, we can support various neural networks in different applications by choosing proper memristor devices, for example PRAM for extremely high-density applications and MRAM for high performance applications. To reduce the computational cost in a neural network composed of a sizable number of memristor devices, matrix-vector multiplication has been studied in memristor crossbar arrays. Although energy efficient neural networks can be implemented in the inference procedure by using matrix-vector multiplication operation, this operation also accumulates cell variations and is not easily achievable for large arrays consisting of considerable numbers of memristor devices. For practical applications beyond research, we should consider the distribution of device parameters and more practical neural network approaches such as binary neural networks are very important for extremely energy efficient computing. Aside from the memory function in practical ANN applications, memristor devices have unique characteristics that emulate some functions of biological neurons, such as STDP and LIF. Because of these characteristics, memristor devices have been studied as a means to accelerate the research of SNNs, which are more bio-plausible and more efficient. Even though memristor devices are very useful for neural network technologies, we should implement them in neural networks by considering not only their advantages but also the technical difficulties. Especially, parameter variation caused by random behaviors during cell operation, variations in fabrication process, and other instabilities are very important factors for establishing memristors in neural networks. The variability and immaturity of memristor devices remains large compared to conventional devices; however, some memristor devices have made apparent progress in

reliability and have achieved commercialization. Considering all aspects, including their pros and cons, we can conclude that MRAM, PRAM, and RRAM are promising technologies for neural networks. Utilizing their characteristics with appropriate applications, we can establish feasible neural networks from the practical approach involving ANNs to the futuristic SNNs in near future.

**References**

1. Chua, L. O., 1971 Memristor—missing circuit element. *IEEE Trans. Circuit Theory* **CT-18**, 507–519.
2. Chua, L. O. and Kang, S. M. 2011 Memristive devices and systems. *Proc. of IEEE* 64, 209–223
3. Chua, L. O. 2011 Resistance switching memories are memristors. *Appl. Phys. A* 102(4), 765–783.
4. Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. 2008 The missing memristor found. *Nature* 453, 80–83
5. Garcia, V. et al., 2009 Giant tunnel electro resistance for non-destructive readout of ferroelectric states *Nature* 460, 81–84.
6. Krzysteczko, P., Münchenberger, J., Schäfers, M., Reiss, G. and Thomas, A., 2012 The Memristive Magnetic Tunnel Junction as a Nanoscopic Synapse-Neuron System *Advanced Materials* 24 (6), 762–766.
7. Arthur H. Edwards, et al., 2015 Reconfigurable Memristive Device Technologies *Proceedings of the IEEE*, 103(7), 1004 – 1033.
8. Bessonov, A. A., et al. 2014 Layered memristive and memcapacitive switches for printable electronics



- Nature Materials 14, 199–204
9. An. Alibart, et.al. 2010 An Organic Nanoparticle Transistor Behaving as a Biological Spiking Synapse Adv. Func. Mater. 20 (2), 330–337.
  10. Zhou, X., et.al., 2013 Phase transition characteristics of Al-Sb phase change materials for phase change memory application Appl. Phys. Lett. 103, 072114.
  11. Kwang-Jin Lee, et.al., 2008 A 90 nm 1.8 V 512 Mb diode-switch PRAM with 266 MB/s read throughput IEEE Journal of Solid-State Circuits 43(1), 150-162
  12. Kyung-Chang Ryoo, Jeong-Hoon Oh, Sunghun Jung, Hongsik Jeong and Byung-Gook Park 2011 Novel U-shape resistive random access memory structure for improving resistive switching characteristics Japanese Journal of Applied Physics 50(4S), 04DD15.
  13. Kim, J.P., et.al., 2011 A 45nm 1Mb embedded STT-MRAM with design techniques to minimize read-disturbance Symposium on VLSI Circuits Dig. Tech. Papers, 296-297.
  14. Dmytro Apalkov, et.al. 2013 Spin-transfer torque magnetic random access memory (STT-MRAM) ACM Journal on Emerging Technologies in Computing Systems 9(2):13,1-35.
  15. Sung Hyun Jo, et.al. 2010 Nanoscale Memristor Device as Synapse in Neuromorphic Systems, Nano Lett., 10 (4), 1297–1301.
  16. Zhongrui Wang, et. al. 2017 Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing Nature Materials 16, 101–108.
  17. Dae-HwanKang, et. al. 2015 Emulation of spike-timing dependent plasticity in nano-scale phase change memory, Neurocomputing, 155, 153-158.
  18. Adrien F. Vincent, et. al. 2015 Spin-Transfer Torque Magnetic Memory as a Stochastic Memristive Synapse for Neuromorphic Systems IEEE Transactions on Biomedical Circuits and Systems 9(2), 166 – 174.
  19. Bart L.M. Happel and Jacob M.J. Murre 1994 Design and evolution of modular neural network architectures Neural Networks ,7, 985-1004.
  20. Daniel S. Levine, 2007 Neural network modeling of emotion Physics of Life Reviews 4(1), 37-63
  21. Stephen K. Reed, 2013 Cognition: theories and applications, 9<sup>th</sup> ed. Wardsworth.
  22. Markus Kuderer, Shilpa Gulati and Wolfram Burgard, 2015 Learning driving styles for autonomous vehicles from demonstration in Proc. of International Conference on Robotics and Automation, 2641-2646.
  23. D. Floreano and F. Mondada, 1998 Evolutionary neuro controllers for autonomous mobile robots Neural Netw. 11 (7-8),1461-1478.
  24. Jatin Borana, 2016 Applications of Artificial Intelligence & Associated Technologies in Proc. of ETEBMS, 64-67.
  25. Jiajun Zhang and Chengqing Zong, 2015 Deep Neural Networks in Machine Translation: An Overview IEEE Intelligent Systems 30(5), 16-25

26. J.A. Noble and D. Boukerroui, 2006 Ultrasound image segmentation: a survey IEEE Trans. Med. imaging 25 (8), 987-1010
27. D.J. Hemanth, C.K.S. Vijila, A.I. Selvakumar and J. Anitha, 2014 Performance improved iteration-free artificial neural networks for abnormal magnetic resonance brain image classification Neurocomputing 130, 98-107.
28. S. Khemakhem and Y. Boujelbène, 2015 Credit risk prediction: a comparative study between discriminant analysis and the neural network approach J. Acc. Manag. Inf. Syst. 14 (1), 60-78.
29. T. Sagara, M. Hagiwara, 2014 Natural language neural network and its application to question-answering system", Neurocomputing 142, 201-208.
30. Navjot Kaur, Amardeep Singh, 2015 Analysis of Vascular Pattern Recognition Using Neural Network, International Journal of Mathematical Sciences and Computing 1(3), 9-19.
31. Michael D Godfrey, David F Hendry, 1993 The computer as von Neumann planned it", IEEE Annals of the History of Computing 15(1), 11-21.
32. Jürgen Schmidhuber, 2015 Deep learning in neural networks: An overview, Neural Networks 61, 85–117.
33. Janardan Misra, Indranil Saha, 2010 Artificial neural networks in hardware: A survey of two decades of progress, Neurocomputing 74, 239-255.
34. Fernando Morgado Dias, Ana Antunes, Alexandre Manuel Mota, 2004 Artificial neural networks: a review of commercial hardware, Engineering Applications of Artificial Intelligence, 17(8), 945-952.
35. M. Prezioso, F. Merrih-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev & D. B. Strukov, 2015 Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521, pp 61–64.
36. S.B. Furber, D.R. Lester, L.A. Plana, J.D. Garside, E. Painkras, S. Temple, A.D. Brown, 2013 Overview of the SpiNNaker system architecture, IEEE Trans. Comput. 62 (12), 2454-2467.
37. J. Schemmel, D. Brüderle, A. Grubl, M. Hock, K. Meier and S. Millner, 2010 A wafer-scale neuromorphic hardware system for large-scale neural modelling", in Proc. of IEEE Int. Symp. Circuits Syst., 1947–1950.
38. Filipp Akopyan, et.al., 2015 True North: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34(10), 1537-1557.
39. Mark Horowitz, 2014 Computing's Energy Problem (and what we can do about it) ISSCC Dig. Tech. Papers, 10-14.
40. F Ponulak, A Kasinski, 2011 Introduction to spiking neural networks: Information processing, learning and applications, Acta Neurobiologiae Experimentalis, 71(4), 409.
41. Xun Wang, Tao Song, Faming Gong, Pan Zheng, 2016 On the Computational Power of Spiking Neural P Systems with Self-Organization, Sci. Rep. 6(1), 27624.
42. Chi-Sang Poon and Kuan Zhou, 2011 Neuromorphic silicon neurons and large-scale neural networks:

- challenges and opportunities, *Front. Neurosci.* 22, 1-3.
43. Yang Zhang, et. al., 2018 Memristor-Based Circuit Design for Multilayer Neural Networks *IEEE Transactions on Circuits and Systems* 65(2), 677 – 686.
  44. Alexander Serb, Johannes Bill, Ali Khiat, Radu Berdan, Robert Legenstein and Themis Prodromakis, 2016 Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses *Nat. Comm.* 7, 12611.
  45. Geoffrey W. Burr et.al., 2016 Neuromorphic computing using non-volatile memory *Advances in Physics:X*, 2(1), 89-124.
  46. Shimeng Yu, 2018 Neuro-Inspired Computing With Emerging Nonvolatile Memories, *Proceedings of the IEEE* 106(2), 260 – 285.
  47. Seunghyun Lee, Joon Sohn, Zizhen Jiang, Hong-Yu Chen, w and H.-S. Philip Wong, 2015 Metal oxide-resistive memory using graphene-edge electrodes, *Nat. Comm.* 6, 8407.
  48. Miao Hu, et. al., 2018 Memristor-Based Analog Computation and Neural Network Classification with a Dot Product Engine *Adv. Mater.* 30, 1705914.
  49. S.-M. Yoon, et. al., 2011 Polymeric ferroelectric and oxide semiconductor-based fully transparent memristor cell *Applied Physics A: Materials Science and Processing* 102(4), 983–990.
  50. Xiaofang Hu, Gang Feng, Shukai Duan, and Lu Liu. 2015 Multilayer RTD-memristor-based cellular neural networks for color image processing *Neurocomputing* 162, 150–162.
  51. Qi Liu, et. al., 2010 Controllable Growth of Nanoscale Conductive Filaments in Solid-Electrolyte Based ReRAM by Using a Metal Nanocrystal Covered Bottom Electrode *ACS Nano* 4(10), 6162–6168.
  52. Sascha Vongehra 2015 Purely Mechanical Memristors: Perfect Massless Memory Resistors, the Missing Perfect Mass Involving Memristor, and Massive Memristive Systems *arXiv:1504.00300v2 [physics.gen-ph]*.
  53. Sascha Vongehr and Xiangkang Meng, 2015 The Missing Memristor has Not been Found, *Sci. Rep.* 5:11657.
  54. Sebastien Couet, et.al. 2016 Oxygen Scavenging by Ta Spacers in Double-MgO Free Layers for Perpendicular Spin-Transfer Torque Magnetic Random-Access Memory, *IEEE Magnetics Letters* 7, 313004.
  55. Chang Y-J, Canizo-Cabrera A, Garcia-Vazquez V, Chang Y-H and Wu T 2013 Effect of Ta thickness on the perpendicular magnetic anisotropy in MgO/CoFeB/Ta/[Co/Pd]<sub>n</sub> structures *J. Appl. Phys* 114, 184303.
  56. K.H. Kim, S.H. Jo, S. Gaba and W. Lu, 2010 Nanoscale resistive memory with intrinsic diode characteristics and long endurance *Appl. Phys. Lett.* 96, 053106.
  57. S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, 2010 Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301.
  58. M. Wuttig and N. Yamada, 2007 Phase-change materials for rewriteable data storage *Nat. Mater.* 6, 824–832.

59. D. Kuzum, R.G.D. Jeyasingh, B. Lee and H.S.P. Wong 2012 Nano electronic programmable synapses based on phase change materials for brain-inspired computing Nano Lett. 12, 2179–2186.
60. Zhongqiang Hu, et al., 2013 Ferroelectric memristor based on Pt/BiFeO<sub>3</sub>/Nb-doped SrTiO<sub>3</sub> heterostructure App. Phys. Lett. 102, 102901, 1-5.
61. Cited from <http://www.fujitsu.com/us/products/devices/semiconductor/memory/fram/>
62. Cited from <http://www.cypress.com/products/f-ram-nonvolatile-ferroelectric-ram>
63. Chanthbouala, A., et al. 2012 A ferroelectric memristor, Nature Materials 11(10), 860–864.
64. Hongsik Jeong and Kinam Kim, 2004 Prospect of Emerging Nonvolatile Memories. In Proc. of MRS on line Proceedings 830, D7.6.
65. Shimeng Yu and Pai-Yu Chen, 2016 Emerging Memory Technologies Recent Trends and Prospects, IEEE Solid-State Circuits Magazine 8(2), 43-56.
66. Jagan Singh Meena et al., 2014 Overview of emerging nonvolatile memory technologies, Nanoscale Research Letters 9, 526, 1-33.
67. Hongsik Jeong 2016 High density PCM(phase change memory) technology. In Proc. of International SoC Design Conference (ISOCC).
68. Duygu Kuzum, Shimeng Yu and H-S PhilipWong 2013 Synaptic electronics: materials, devices and applications Nanotechnology 24, 382001.
69. Y. J. Song, et.al., 2016 Highly Functional and Reliable 8Mb STT-MRAM Embedded in 28nm Logic, IEDM Dig. Tech. Papers, 10.2.,1–4.
70. Quantan Wu, et.al., 2018 Full imitation of synaptic meta plasticity based on memristor devices, Nanoscale 10, 5875-5881.
71. Cited from [https://ark.intel.com/products/97544/Intel-Optane-Memory-Series-16GB-M\\_2-80mm-PCIe-3\\_0-20nm-3D-Xpoint](https://ark.intel.com/products/97544/Intel-Optane-Memory-Series-16GB-M_2-80mm-PCIe-3_0-20nm-3D-Xpoint)
72. Cited from Samsung's private data sheet
73. Cited from <https://industrial.panasonic.com/ww/products/semiconductors/microcomputers/>
74. Cited from <http://www.fujitsu.com/global/documents/products/devices/semiconductor/memory/reram/>
75. Dongku Kang, et. al., 2016 256Gb 3b/Cell V-NAND Flash Memory with 48 Stacked WL Layers ISSCC Dig. Tech. Papers, 130-132.
76. Ayushi and Rajesh Mehra 2017 Energy Efficient FinFET based SRAM Design in 22-Nanometer Technology In Proc. of SARC International Conference.
77. S.-W. Chung, et.al., 4Gbit density STT-MRAM using perpendicular MTJ realized with compact cell structure IEDM Dig. Tech. Papers, 659-662.
78. Sabpreet Bhatti, Rachid Sbiba, Atsufumi Hirohata, Hideo Ohno, Shunsuke Fukami and S.N. Piramanayagam, 2017 Spintronics based random access memory: a review, Materials Today 20(9), 530-548.

79. Shoji Ikeda, et. al., 2007 Magnetic Tunnel Junctions for Spintronic Memories and Beyond, IEEE Transactions on electron devices 54(5), 991-1002.
80. Ikeda, S.; Miura, K.; Yamamoto, H.; Mizunuma, K.; Gan, H. D.; Endo, M.; Kanai, S.; Hayakawa, J.; Matsukura, F.; Ohno, H. Nat. Mater. 2010, 9 (9), 721–724.
81. B. Dieny and M. Chshiev, 2017 Perpendicular magnetic anisotropy at transition metal/oxide interfaces and applications Rev. Mod. Phys. 89, 025008.
82. H. Zhao et al., 2012 Sub-200 ps spin transfer torque switching in in-plane magnetic tunnel junctions with interface perpendicular anisotropy, J. Phys. D: Appl. Phys. 45, 025001.
83. Ju Hyun Kim, et. al., 2014 Verification on the extreme scalability of STT-MRAM without loss of thermal stability below 15 nm MTJ cell, Symposium on VLSI Dig. Tech. Papers., 1-2.
84. M. Hosomi, et al., 2005 A Novel Nonvolatile Memory with Spin Torque Transfer Magnetization Switching: Spin-RAM, IEDM Dig. Tech. Papers.
85. Huai, Y. et al, 2015, PMTJ driven STT MRAM with 300nm process In Proc. of IEEE Magnetics Conference (INTERMAG).
86. Hu, G. et al, 2015 STT-MRAM with double magnetic tunnel junctions, IEDM Dig. Tech. Papers, 26-30.
87. Noguchi, H., et al, 2015 A 3.3 ns-access-time 71.2 $\mu$ W/MHz 1Mb embedded STT-MRAM using physically eliminated read-disturb scheme and normally-off memory architecture ISSCC Dig. Tech. Papers, 1-3.
88. Grezes C, et. al., 2016 Ultra-low switching energy and scaling in electric-field-controlled nanoscale magnetic tunnel junctions with high resistance-area product. Appl. Phys. Lett. 108, 012403.
89. Sungwoo Park, et.al., 2017 Improved Etch Characteristics of Magnetic Tunneling Junction Materials by Using Helium ECS Journal of Solid State Science and Technology 6(9), 148-154.
90. Matthias Wuttig and Noboru YaMada, 2007 Phase-change materials for rewriteable data storage, Nature Materials 6, 824-832.
91. M.-C. Jung, et.al., 2007 Ge nitride formation in N-doped amorphous Ge<sub>2</sub>Sb<sub>2</sub>Te<sub>5</sub> Appl. Phys. Lett. 91, 083514.
92. Konstantin B.Borisenko et. al., 2011 Understanding atomic structures of amorphous C-doped Ge<sub>2</sub>Sb<sub>2</sub>Te<sub>5</sub> phase-change memory materials, Acta Materialia 59(11), 4335-4342.
93. T. Ohyanagi et al., 2013 Charge-injection phase change memory with high quality GeTe/Sb<sub>2</sub>Te<sub>3</sub> superlattice featuring 70- $\mu$ A reset, 10-ns set and 100 m endurance cycles operations, IEDM Tech. Dig. Papers, 30.5.1–30.5.4.
94. M. Tai et al., 2014 1T-1R pillar-type topological-switching random access memory (TRAM) and data retention of GeTe/Sb<sub>2</sub>Te<sub>3</sub> super-lattice films Symposium on VLSI Technology Dig. Tech. Papers, 1-2.
95. Jinil Lee, et.al., 2011 Scalable High-Performance Phase-Change Memory Employing CVD GeBiTe IEEE Electron Device Letters 32(8), 1113 – 1115.
96. Feng Rao, et.al., 2017 Reducing the stochasticity of crystal nucleation to enable sub-nanosecond

- memory writing, *Science* 9, 3212.
97. F. Pellizzer, et. al., 2004 Novel pTrench Phase-Change Memory Cell for Embedded and Stand-Alone Non-Volatile Memory Applications Symposium on VLSI Tech. Dig. Tech. Papers, 18-19.
  98. Feng Xiong, et.al., 2013 Self-Aligned Nanotube–Nanowire Phase Change Memory *Nano Letters* 13 (2), 464-469.
  99. Milos Stanisavljevic, et.al., 2016 Demonstration of Reliable Triple-Level-Cell (TLC) Phase-Change Memory In Proc. of IEEE 8th International Memory Workshop (IMW)
  100. H. Y. Cheng et al., 2015 Novel fast-switching and high-data retention phase-change memory based on new Ga-Sb-Ge material IEDM Tech. Dig. Papers, 3.5.1-3.5.4.
  101. T.-Y. Liu, et. al., 2014 A 130.7mm<sup>2</sup> 2-Layer 32Gb ReRAM Memory Device in 24nm Technology *IEEE Journal of Solid-State Circuits* 49(1), 140 – 153.
  102. K.-S. Li, et al. 2014 Utilizing Sub-5 nm sidewall electrode technology for atomic-scale resistive memory fabrication Symposium on VLSI Technology Dig. Tech. Papers, 1-2.
  103. Jen-Chieh Liu, Chung-Wei Hsu, I-Ting Wang, and Tuo-Hung Hou, 2015 Categorization of Multilevel-Cell Storage-Class Memory: An RRAM Example *IEEE Transactions on Electron Devices* 62(8), 2510 – 2516.
  104. E. Cha, et al., 2013 Nanoscale (~10nm) 3D vertical ReRAM and NbO<sub>2</sub> threshold selector with TiN electrode, IEDM Tech. Dig. Papers 10.5.1-10.5.4.
  105. H. Y. Lee, et. al., 2010 Evidence and solution of Over-RESET Problem for HfO<sub>x</sub> Based Resistive Memory with Sub-ns Switching Speed and High Endurance, IEDM Dig. Tech. Papers, 19.7.1 - 19.7.4.
  106. W. C. Shen, et al., 2012 High-K Metal Gate Contact RRAM (CRRAM) in Pure 28nm CMOS Logic Process, IEDM Dig. Tech. Papers, 31.6.1 - 31.6.4.
  107. A. Kawahara, et. al., 2012 An 8Mb Multi-Layered Cross-Point ReRAM Macro with 443MB/s Write Throughput ISSCC Tech. Dig. Papers, 432 – 434.
  108. C. -W. Hsu et. al., 2013 Self-Rectifying Bipolar TaO<sub>x</sub>/TiO<sub>2</sub> RRAM with Superior Endurance over 10<sup>12</sup> Cycles for 3D High-Density Storage-Class Memory, Symposium on VLSI Technology Dig. Tech. Papers, 166-167.
  109. M. Wang, et. al., 2010 A Novel Cu<sub>x</sub>Si<sub>y</sub>O Resistive Memory in Logic Technology with Excellent Data Retention and Resistance Distribution for Embedded Applications Symposium on VLSI Technology Dig. Tech. Papers, 89-90.
  110. Cited from <https://nano.stanford.edu/stanford-memory-trends>
  111. S. G. Hu, S. Y. Wu, W.W. Jia, Q. Yu, L. J. Deng, Y. Q. Fu, Y. Liu, and T. P. Chen 2014 Review of Nanostructured Resistive Switching Memristor and Its Applications *Nanoscience and Nanotechnology Letters* 6, 729–757.
  112. J. Joshua Yang, Dmitri B. Strukov and Duncan R. Stewart, 2013 Memristive devices for computing *Nature Nanotechnology* 8, 13-24.
  113. Daniele Ielmini, Federico Nardi, and Carlo Cagli, 2010 Resistance-dependent amplitude of random

- telegraph-signal noise in resistive switching memories *App. Phys. Lett.* 96, 053503.
114. Stefano Ambrogio, et.al., 2014 Statistical Fluctuations in HfOx Resistive-Switching Memory: Part I - Set/Reset Variability, *IEEE Trans. On Elec. Devices* 61 (8), 2912-2919.
  115. Stefano Ambrogio, et.al., 2014 Statistical Fluctuations in HfOx Resistive-Switching Memory: Part II—Random Telegraph Noise. *IEEE Trans. On Elec. Devices* 61 (8), 2920-2927.
  116. Alessandro Fumarola, et. al., 2018 Bidirectional Non-Filamentary RRAM as an Analog Neuromorphic Synapse, Part II: Impact of Al/Mo/Pr<sub>0.7</sub>Ca<sub>0.3</sub>MnO<sub>3</sub> Device Characteristics on Neural Network Training Accuracy *J. of Electron devices society* 6, 169-178.
  117. M.D. Godfrey and D.F. Hendry 1993 The computer as von Neumann planned it, *IEEE annals of the History of Computing*, 15(1), 11-21.
  118. Gordon E. Moore, 1995 Lithography and the Future of Moore's Law, in *Proc. of SPIE* Vol. 2438.
  119. P. Malik, 2013 Governing Big Data: Principles and practices, *IBM J. of Res. and Dev.*, 57, 1-13.
  120. A. M. Turing, 1936 On Computable Real Numbers, with an Application to the Entscheidungs problem, *Proceedings of the London Mathematical Society*, 42(2), 230-265.
  121. John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon, 2006 A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, *AI Magazine* 27(4), 12-14.
  122. Mark Fischetti, 2011 Computers vs. Brains *Scientific American* 305(5), 104.
  123. David Silver, et.al., 2016 Mastering the game of Go with deep neural networks and tree search, *Nature* 529, 484–489.
  124. David Silver, et.al., 2017 Mastering the game of Go without human knowledge, *Nature* 550, 354–359.
  125. Cited from <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai>.
  126. Koushal Kumar, Gour Sundar and Mitra Thakur, 2012 Advanced Applications of Neural Networks and Artificial Intelligence: A Review *Int. J. of Information Technology and Computer Science* 6, 57-68
  127. Javier Bajo and Juan M. Corchado, 2018 Neural networks in distributed computing and artificial intelligence, *Neurocomputing* 272(10), 1-2.
  128. Jure Zbontar and Yann LeCun, 2016 Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches *Journal of Machine Learning Research* 17(65), 1-32.
  129. B. Widrow and M.A. Lehr, 1990 30 years of adaptive neural networks: perceptron, Madaline, and back propagation *Proc. of the IEEE* 78(9), 1415 – 1442.
  130. A.K. Jain, Jianchang Mao and K.M. Mohiuddin, 1996 Artificial neural networks: a tutorial, *Computer* 29(3), 31 – 44.
  131. Kurt Hornik, Maxwell Stinchcombe and Halbert White, 1989 Multilayer feedforward networks are universal approximators *2(5)*, 359-366.
  132. Hinton, G. E., Osindero, S., and Teh, Y. W., 2006 A fast learning algorithm for deep belief nets, *Neural*

Computation 18, 1527-1554.

133. W.Zhao, R.Chellappa,P. J.Phillips, and A.Rosenfeld, 2003 Face recognition: a literature survey ACM Comput.Surv.(CSUR) 35(4), 399–458.
134. Z.L. Sun, H, Wang,W.S, Lau,G.Seet and D.W.Wang, 2014 Application of BW-ELM model on traffic sign recognition, Neurocomputing 128, 153–159.
135. Pooja Yadav and Nidhika Yadav, 2015 Handwriting Recognition System- A Review, International Journal of Computer Applications 114, 36-40.
136. Joonatas Wehrmann, Willian Becker, Henry E. L. Cagnini and Rodrigo C. Barros, 2017 A character-based convolutional neural network for language-agnostic Twitter sentiment analysis, in Proc. of International Joint Conference on Neural Networks.
137. Joseph D. Prusa and Taghi M. Khoshgoftaar, 2017 Improving deep neural network design with new text data representations, Journal of Big Data 4:7, 1-16.
138. Hao Zhou, Yue Zhang, Chuan Cheng, Shujian Huang, Xinyu Dai and Jiajun Chen, 2017 A Neural Probabilistic Structured-Prediction Method for Transition-Based Natural Language Processing Journal of Artificial Intelligence Research 58, 703-729.
139. I. Kaastra and M.Boyd, 1996 Designing a neural network for forecasting financial and economic time series, Neurocomputing10(3), 215–236.
140. T.G. Barbounis and J.B. Theocharis, 2006 Locally recurrent neural networks for long-term wind speed and power prediction Neurocomputing 69, 466-496.
141. W.Z.Lu, H.Y.Fan and S.M.Lo, 2003 Application of evolutionary neural network method in predicting pollutant levels in downtown area of Hong Kong, Neurocomputing 51, 387–400.
142. Li Deng, Geoffrey Hinton and Brian Kingsbury, 2013 New types of deep neural network learning for speech recognition and related applications: An overview in Proc. of IEEE International Conference on Speech and Signal Processing (ICASSP), 8599-8603.
143. Geoffrey Hinton, et. al., 2012 Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups, : IEEE Signal Processing Magazine **29**(6), 82 – 97.
144. J.A.Noble and D.Boukerroui, 2006 Ultrasound image segmentation: a survey IEEE Trans. Med. imaging 25(8), 987–1010.
145. M.G. Reese, 2001 Application of a time-delay neural network to promoter annotation in the drosophila melanogaster genome, Comput. Chem. 26(1), 51–56.
146. J.D. Wulfschle, L.A.Liotta and E.F.Petricoin, 2003 Proteomic applications for the early detection of cancer, Nat. Rev. Cancer 3(4), 267–275.
147. ShichunYang, Yaoguang Cao, Zhaoxia Peng, GuoguangWen and Konghui Guo, 2017 Distributed formation control of nonholonomic autonomous vehicle via RBF neural network, Mechanical Systems and Signal Processing 87(B), 81-95.
148. Jian Gao, Alison A. Proctor, Yang Shi and Colin Bradley, 2016 Hierarchical Model Predictive Image-Based Visual Servoing of Underwater Vehicles With Adaptive Neural Network Dynamic Control, IEEE Transactions on Cybernetics 46(10), 2323 – 2334.



149. H.N. Nguyen, J.Zhou and H.J.Kang, 2015 A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network, *Neurocomputing* 151, 996–1005.
150. Tong Wang, Huijun Gao and Jianbin Qiu, 2016 A Combined Adaptive Neural Network and Nonlinear Model Predictive Control for Multirate Networked Industrial Process Control, *IEEE Transactions on Neural Networks and Learning Systems* 27(2), 416 – 425.
151. Mehmet Turanad, Yasin Almalioglu, Helder Araujo, Ender Konukoglu and Metin Sittia, Deep EndoVO: A recurrent convolutional neural network (RCNN) based visual odometry approach for endoscopic capsule robots, *Neurocomputing* in Press.
152. Wei He, Amoateng, Ofori David, Zhao Yin and Changyin Sun, 2016 Neural Network Control of a Robotic Manipulator With Input Dead zone and Output Constraint, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46(6), 759 – 770.
153. Cited from <http://www.asimovinstitute.org/neural-network-zoo>.
154. Ankush Acharyya, et. al., 2013 Handwritten Word Recognition Using MLP based Classifier: A Holistic Approach, *International Journal of Computer Science* 10(2), 422-427.
155. Anita Pal and Dayashankar Singh, 2010 Handwritten English Character Recognition Using Neural Network, *International Journal of Computer Science & Communication* 1(2), 141-144.
156. J.I. Peláez, J.M. Doña, J.F. Fornari and G Serra, 2014 Ischemia classification via ECG using MLP neural networks, *International Journal of Computational Intelligence Systems* 7(2), 344-352.
157. Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, 2012 ImageNet Classification with Deep Convolutional Neural Networks, in *Proc. of NIPS*.
158. S. Lawrence, C.L. Giles, Ah Chung Tsoi and A.D. Back, 1997 Face recognition: a convolutional neural-network approach, *IEEE Transactions on Neural Networks* 8(1), 98-113.
159. Masakazu Matsugu, Katsuhiko Mori, Yusuke Mitari and Yuji Kaneda, 2003 Subject independent facial expression recognition with robust face detection using a convolutional neural network, *Neural Networks* 16(5–6), 555-559.
160. Shuiwang Ji, Wei Xu, Ming Yang and Kai Yu, 2013 3D Convolutional Neural Networks for Human Action Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(1), 221 – 231.
161. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke and Alexander A. Alemi, 2017 Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, in *Proc. of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*.
162. Ken-ichi Funahashi and Yuichi Nakamura, 1993 Approximation of dynamical systems by continuous time recurrent neural networks, *Neural Networks* 6(6), 801-806.
163. Mantas Lukoševičius and Herbert Jaeger, 2009 Reservoir computing approaches to recurrent neural network training, *Computer Science Review* 3(3), 127-149.
164. Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký and Sanjeev Khudanpur, 2011 Extensions of recurrent neural network language model, In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5528-5531.

165. Yunong Zhang and S.S. Ge, 2005 Design and analysis of a general recurrent neural network model for time-varying matrix inversion, *IEEE Transactions on Neural Networks* 16(6), 1477 – 1490.
166. N. F. Hardy and Dean V. Buonomano, 2017 Encoding Time in Feedforward Trajectories of a Recurrent Neural Network Model, *Neural Computation*, 1-19.
167. Wolfgang Maass, 1997 Networks of spiking neurons: The third generation of neural network models, *Neural Networks* 10(9), 1659-1671.
168. Dan Goodman<sup>1</sup> and Romain Brette, 2008 Brian: A Simulator for Spiking Neural Networks in Python *Frontiers in Neuroinformatics* 2, 5, 1-10.
169. Jayra Moorkanikara, et.al., 2009 A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors, *Neural Networks* 22(5–6), 791-800.
170. Nikola K.Kasabov, 2014 NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, *Neural Networks* 52, 62-76.
171. Damien Querlioz, Olivier Bichler, Philippe Dollfus and Christian Gamrat, 2013 Immunity to Device Variations in a Spiking Neural Network With Memristive Nanodevices, *IEEE Transactions on Nanotechnology* 12(3), 288 – 295.
172. Romain Brette, et.al., 2007 Simulation of networks of spiking neurons: A review of tools and strategies, *Journal of Computational Neuroscience* 23(3), 349–398.
173. Giacomo Indiveri and Shih-Chii Liu, Memory and, 2015 Information Processing in Neuromorphic Systems, *Proceedings of the IEEE* 103(8), 1379-1397.
174. M. Hill and M. Marty, 2008 Amdahl's law in the multicore era, *IEEE Computer* 41(7), 33–38.
175. A. Cassidy and A. Andreou, 2012 Beyond Amdahl's law: An objective function that links multiprocessor performance gains to delay and energy, *IEEE Trans. Comput.*, 61(8), 1110–1126.
176. R. Brette and D. Goodman, 2012 Simulating spiking neural networks on GPU, *Network, Comput. Neural Syst.* 23(4), 167–182.
177. D. Neil and S.-C. Liu, 2014 Minitaur, an event-driven FPGA based spiking network accelerator *IEEE Trans. VLSI Syst.*, 22(12), 2621–2628.
178. C. Farabet, et. al., 2011 Neuflow: A runtime reconfigurable dataflow processor for vision, in *Proc. of IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 109–116.
179. A. Majumdar, S. Cadambi, M. Becchi, S. Chakradhar, and H. Graf, 2012 A massively parallel, energy efficient programmable accelerator for learning and classification, *ACM Trans. Architect. Code Optim.* 9(1), 6:1–6:30.
180. S. Furber, F. Galluppi, S. Temple, and L. Plana, 2014 The SpiNNaker project *Proc. of IEEE* 102(5), 652–665.
181. P. A. Merolla, et. al., 2014 A million spiking-neuron integrated circuit with a scalable communication network and interface, *Science* 345(6197), 668–673.
182. B. V. Benjamin, et. al., 2014 Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations, *Proc. of IEEE* 102(5), 699–716.

183. J. Schemmel, D. Brüderle, K. Meier, and B. Ostendorf, 2007 Modeling synaptic plasticity within networks of highly accelerated I&F neurons in Proc. of IEEE Int. Symp. Circuits Syst., 3367–3370.
184. N. Qiao et al., 2015 A re-configurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses, Front. Neurosci. 9(141).
185. Bo Zhang, Luping Shi and Sen Song, 2016 Creating more intelligent robots through brain inspired computing, special supplement: Brain Inspired Robotics Science 354(6318), 4-9.
186. Shubham Gupta and M. Rajasekhara Babu, 2011 Performance Analysis of GPU compared to Single core and Multi-core CPU for Natural Language Applications Int. J. of Advanced Comp. Sci. and Applications 2(5), 50-53.
187. Tao Luo, et.al., 2017 DaDianNao: A Neural Network Supercomputer, IEEE Transactions on Computers 66(1), 73 – 88.
188. Cited from [http://isscc.org/wp-content/uploads/sites/10/2017/05/ISSCC2017\\_PressKit.pdf](http://isscc.org/wp-content/uploads/sites/10/2017/05/ISSCC2017_PressKit.pdf)
189. Tarek M. Taha, Raqibul Hasan, Chris Yakopcic and Mark R. McLean, 2013 Exploring the design space of specialized multicore neural processors, in Proc. The International Joint Conference on Neural Networks (IJCNN).
190. Ben Varkey Benjamin, et.al., 2014 Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations Proceedings of the IEEE, 102(5), pp. 699 – 716.
191. Yunji Chen, et.al., 2014 DaDianNao: A Machine-Learning Supercomputer, 47th Annual IEEE/ACM International Symposium on Microarchitecture, pp.609-622.
192. Johannes Schemmel, Johannes Fieres and Karlheinz Meier, 2008 Wafer-Scale Integration of Analog Neural Network, in Proc. of IEEE International Joint Conference on Neural Networks(IJCNN).
193. W. -H. Chen, et al., 2018 A 65nm 1Mb Nonvolatile Computing-in-Memory ReRAM Macro with sub-16ns Multiply-and-Accumulate for Binary DNN AI Edge Processor, ISSCC, Dig. Tech. Papers, 494 - 496.
194. Dongwook Kim, Youjip Won, Jaehyuk Cha, Sungroh Yoon, Senior Member, Jongmoo Choi and Sooyong Kang, 2016 Exploiting Compression-Induced Internal Fragmentation for Power-Off Recovery in SSD, IEEE Transactions on computers 65(6), 1720-1733.
195. Hyun-Seob Lee, Sangwon Park and Dong-Ho Lee, 2013 RMSS: an efficient recovery management scheme on NAND flash memory based solid state disk, IEEE Transactions on Consumer Electronics 59(1), 107-112.
196. Koichi Shirahata, Yasumoto Tomita, Atsushi Ike, 2016 Memory reduction method for deep neural network training, in Proc. of IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP).
197. Bradley McDanel, Surat Teerapittayanon and H.T. Kung 2017 Embedded Binarized Neural Networks arXiv:1709.02260v1.
198. Yu Wang, Tianqi Tang, Lixue Xia, Boxun Li, Peng Gu, Hai Li, Yuan Xie and Huazhong Yang, 2015 Energy Efficient RRAM Spiking Neural Network for Real Time Classification, in Proc. of the 25th edition on Great Lakes Symposium on VLSI, 189-194.

199. Lixue Xia, et. al., 2016 Technological Exploration of RRAM Crossbar Array for Matrix-Vector Multiplication, *J. of Computer Science and Technology* 31(1), 3–19.
200. Wu Y, Yu S, Wong H S P, Chen Y S, Lee H Y, Wang S M, Gu P Y, Chen F and Tsai M J, 2012 AlOx-based resistive switching device with gradual resistance modulation for neuromorphic device application, in *Proc. of 4th IEEE Int. Memory Workshop (IMW)*, 1–4.
201. Eduardo Perezi, et. al., 2017 Impact of the Incremental Programming Algorithm on the Filament Conduction in HfO<sub>2</sub>-Based RRAM Arrays, *IEEE J. of Electron Device Society* 5(1), 64-68.
202. Emmanuelle J Merced-Grafals, et. al., 2016 Repeatable, accurate, and high speed multilevel programming of memristor 1T1R arrays for power efficient analog computing applications, *Nanotechnology* 27, 365202, 1-9.
203. Patrick M. Sheridan, Fuxi Cai, Chao Du, Wen Ma, Zhengya Zhang and Wei D. Lu, 2017 Sparse coding with memristor networks *Nature Nanotechnology* 12, 784-790.
204. Lixue Xia, et. al., 2016 MNSIM: Simulation Platform for Memristor-based Neuromorphic Computing System, in *Proc. of Design, Automation & Test in Europe Conference & Exhibition*, 469-474
205. Raqibul Hasan, Tarek M. Taha, Chris Yakopcic and David J. Mountain, 2016 High Throughput Neural Network based Embedded Streaming Multicore Processors, in *Proc. of IEEE International Conference on Rebooting Computing (ICRC)*.
206. Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv and Yoshua Bengio, 2016 Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or -1, *arXiv:1602.02830 v3*.
207. Shuiming Zhong, Xiaoqin Zeng, Shengli Wu and Lixin Han, 2012 Sensitivity-Based Adaptive Learning Rules for Binary Feedforward Neural Networks, *IEEE Transactions on Neural Networks and Learning Systems* 23(3), 480 – 491.
208. Renzo Andri, Lukas Cavigelli, Davide Rossi and Luca Benini, 2018 YodaNN: An Architecture for Ultralow Power Binary-Weight CNN Acceleration, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37(1), 48 – 60.
209. Leibin Ni, Zichuan Liu, Hao Yu and Rajiv V. Joshi, 2017 An Energy-Efficient Digital ReRAM-Crossbar-Based CNN With Bitwise Parallelism, *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* 3, 37 – 46.
210. Shimeng Yu, Zhiwei Li, Paj-Yu Chen, Huaqiang Wu, Bin Gao, Deli Wang, Wei Wu and He Qian, 2016 Binary neural network with 16 Mb RRAM macro chip for classification and online training *IEDM Dig. of Tech. Papers*, 416-419.
211. D. Garvin, et.al., 2015 On the Impact of OxRAM-based Synapses Variability on Convolutional Neural Networks Performance in *Proc. of IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*.
212. Johannes Bill and Robert Legenstein, 2014 A compound memristive synapse model for statistical learning through STDP in spiking neural networks, *Frontiers in Neuroscience* 8, Article412, 1-18.
213. Yu Cai, Onur Mutlu, Erich F. Haratsch and Ken Mai, 2013 Program interference in MLC NAND flash memory: Characterization, modeling, and mitigation, *IEEE 31st International Conference on Computer*

- Design (ICCD), 123-130.
214. M. Suri, et.al., 2012 CBRAM Devices as Binary Synapses for Low-Power Stochastic Neuromorphic Systems: Auditory (Cochlea) and Visual (Retina) Cognitive Processing Applications, IEDM Dig. of Tech. Papers, 236-238.
  215. A.A.Bagheri-Soulla M.B and Ghaznavi-Ghouschi, 2017 An RRAM-based MLC design approach Microelectronics Journal, 64, 9-18.
  216. Sangbum Kim, et.al., 2016 A Phase Change Memory Cell With Metal Nitride Liner as a Resistance Stabilizer to Reduce Read Current Noise for MLC Optimization IEEE Transactions on Electron Devices 99, 1-6.
  217. Farnood Merrikh Bayat, Mirko Prezioso, Bhaswar Chakrabarti, Irina Kataeva and Dmitri Strukov 2017 Memristor-based perceptron classifier: Increasing complexity and coping with imperfect hardware in Proc. of IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 549 – 554.
  218. Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv and Yoshua Bengio, 2016 Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations, arXiv:1609.07061 [cs.NE].
  219. Jinmook Lee, et. al., 2018 UNPU: A 50.6TOPS/W unified deep neural network accelerator with 1b-to-16b fully-variable weight bit-precision, ISSCC Dig. of Tech. Papers, 218 – 220.
  220. Felix Juefei-Xu, Vishnu Naresh Boddeti and Marios Savvides, 2017 Local Binary Convolutional Neural Networks, in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4284 – 4293.
  221. Matthieu Courbariaux, Yoshua Bengio and Jean-Pierre David, 2015 BinaryConnect: Training Deep Neural Networks with binary weights during propagations, in Proc. of Advances in Neural Information Processing Systems(NIPS 2015).
  222. Kota Ando, et. al., 2018 BRein Memory: A Single-Chip Binary/Ternary Reconfigurable in-Memory Deep Neural Network Accelerator Achieving 1.4 TOPS at 0.6 W, IEEE Journal of Solid-State Circuits 53(4), 983 – 994.
  223. Norman P. Jouppi, Cliff Young, Nishant Patil, and David Patterson, 2018 Motivation for and Evaluation of the First Tensor Processing Unit IEEE Micro 38(3), 10-19.
  224. Cited from <https://www.blog.google/topics/google-cloud/google-cloud-offer-tpus-machine-learning/>
  225. Cited from <https://www.nvidia.com/en-us/deep-learning-ai/inference-platform/>
  226. Mohammad Rastegari, Vicente Ordonez, Joseph Redmon and Ali Farhadi, 2016 XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks ECCV 2016, 525-542.
  227. Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer and Shih-Chii Liu, 2017 Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification, Front. Neurosci., 07(11), 686 pp1-12.
  228. Wolfgang Maass and Thomas Natschl"ager, 1997 Networks of spiking neurons can emulate arbitrary Hopfield nets in temporal coding Network: Comput. Neural Syst. 8, 355–371

229. Szatmáry B and Izhikevich EM, 2010 Spike-Timing Theory of Working Memory. *PLoS Comput. Biol.* 6(8): e1000879.
230. Nikola Kasabov, Kshitij Dhoblea, Nuttapod Nuntalid and Giacomo Indiveri, 2013 Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition, *Neural Networks* 41, 188–201.
231. Romain Brette, 2006 Exact Simulation of Integrate-and-Fire Models with Synaptic Conductances, *Neural Computation* 18, 2004–2027.
232. Nicolas Brunel, Vincent Hakim and Magnus JE Richardson, 2014 Single neuron dynamics and computation, *Current Opinion in Neurobiology* 25, 149–155.
233. Bi. G and Poo M M, 1998 Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type *J. Neurosci.* 18, 10464–72.
234. Song S and Abbott L F, 2001 Cortical development and remapping through spike timing-dependent plasticity, *Neuron* 32, 339–50.
235. Song S, Miller K D and Abbott L F, 2000 Competitive Hebbian learning through spike-timing-dependent synaptic plasticity, *Nature Neurosci.* 3, 919–26.
236. G. Indiveri, F. Corradi, and N. Qiao, 2015 Neuromorphic architectures for spiking deep neural networks, *IEDM Dig. Tech. Papers*, 68–71.
237. N. Qiao and G. Indiveri, 2016 Scaling mixed-signal neuromorphic processors to 28 nm FD-SOI technologies,” in *Proc. of IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, 552–555.
238. K. Seo, I. Kim, S. Jung, M. Jo, S. Park, J. Park, J. Shin, K. P. Biju, J. Kong, K. Lee, B. Lee, and H. Hwang, 2011 Analog memory and spike timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device, *Nanotechnology*, 22(25), 254023.
239. Nan Zheng, et. al., 2018 Learning in Memristor Crossbar-Based Spiking Neural Networks Through Modulation of Weight-Dependent Spike-Timing-Dependent Plasticity *IEEE Transactions on Nanotechnology* 17(3), 520 - 532.
240. E. Covi, S. Brivio, A. Serb, T. Prodromakis, M. Fanciulli, and S. Spiga, 2016 Analog memristive synapse in spiking networks implementing unsupervised learning, *Frontiers Neurosci.*, 10, 482.
241. S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H.-S. P. Wong, 2013 A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation, *Adv. Mater.*, 25(12), 1774–1779.
242. Zidong Du, et. al., 2015 Neuromorphic Accelerators: A Comparison Between Neuroscience and Machine-Learning Approaches *MICRO-48*, 494-507.
243. B. Belhadj, A. Joubert, Z. Li, R. Heliot, and O. Temam, 2013 Continuous Real-World Inputs Can Open Up Alternative Accelerator Designs, in *International Symposium on Computer Architecture*.
244. Hebb D.O, 1950 Organization of behavior. New York: Wiley, 1949, pp. 335, *J. Clin. Psychol.* 6, 307.
245. Stefano Ambrogio, Nicola Ciocchini, Mario Laudato, Valerio Milo, Agostino Pirovano, Paolo Fantini and Daniele Ielmini, 2016 Unsupervised Learning by Spike Timing Dependent Plasticity in Phase Change Memory (PCM) Synapses, *Front. Neurosci.* 10, 56.

246. Duygu Kuzum, Rakesh G. D. Jeyasingh, Byoungil Lee, and H.-S. Philip Wong, 2012 Nanoelectronic Programmable Synapses Based on Phase Change Materials for Brain-Inspired Computing, *Nano Lett.* 12(5), 2179–2186.
247. Olivier Bichler, Manan Suri, Damien Querlioz, Member, IEEE, Dominique Vuillaume, Barbara DeSalvo and Christian Gamrat, 2012 Visual Pattern Extraction Using Energy-Efficient 2-PCM Synapse Neuromorphic Architecture, *IEEE Transactions on Elec. Dev.* 59(8), 2206-2214.
248. Abhronil Sengupta, Zubair Al Azim, Xuanyao Fong, and Kaushik Roy, 2015 Spin-orbit torque induced spike-timing dependent plasticity, *Appl. Phys. Lett.* 106(09), 093704
249. Stefano Ambrogio, et. al., 2016 Neuromorphic Learning and Recognition With One-Transistor-One-Resistor Synapses and Bistable Metal Oxide RRAM, *IEEE Transactions on Elec. Dev.* 63(4), 1508-1515.
250. S. Lashkare, N. Panwar, P. Kumbhare, B. Das, and U. Ganguly, 2017 PCMO-Based RRAM and NPN Bipolar Selector as Synapse for Energy Efficient STDP, *IEEE Elec. Dev. Lett.* 38(9), 1212-1215.
251. X. B. Yan, J. H. Zhao, S. Liu, Z. Y. Zhou, Q. Liu, J. S. Chen and X. Y. Liu, 2017 Memristor with Ag-Cluster-Doped TiO<sub>2</sub> Films as Artificial Synapse for Neuroinspired Computing *Adv. Funct. Mater.*, 1705320.
252. Choi H, Jung H, Lee J, Yoon J, Park J, Seong D, Lee W, Hasan M, Jung G Y and Hwang H 2009 An electrically modifiable synapse array of resistive switching memory *Nanotechnology* 20 345201.
253. Seo K, Kim I, Jung S, Jo M, Park S, Park J, Shin J, Biju K P, Kong J and Lee K 2011 Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device *Nanotechnology* 22 254023.
254. Yang R, Terabe K, Liu G, Tsuruoka T, Hasegawa T, Gimzewski J K and Aono M 2012 On-demand nanodevice with electrical and neuromorphic multifunction realized by local ion migration *ACS Nano* 6, 9515–9521.
255. Li-Wei Feng, et.al., 2010 Improvement of resistance switching characteristics in a thin FeOx/FeOx transition layer of TiN/SiO<sub>2</sub>/FeOx/FePtTiN/SiO<sub>2</sub>/FeOx/FePt structure by rapid annealing *Appl. Phys. Lett.* 96, 222108.
256. Yu S, Wu Y, Jeyasingh R, Kuzum D and Wong H P 2011 An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation *IEEE Trans. Electron Devices* 58 2729–37
257. Chao X, Biao N, Long Z, Long-Hui Z, Yong-Qiang Y, Xian-He W, Qun-Ling F, Lin-Bao L and Yu-Cheng W, 2013 High-performance nonvolatile Al/AlO<sub>x</sub>/CdTe: Sb nanowire memory device *Nanotechnology* 24(35):355203.
258. 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 *IEDM Dig. Tech. Papers*, 10.4.1–4.
259. Park S et al 2012 RRAM-based synapse for neuromorphic system with pattern recognition function *IEDM Dig. Tech. Papers*, 10.2.1–4.
260. Geoffrey W. Burr 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*

- Transactions on Electron Devices 62(11), 3498-3507.
261. Yi Wu, et.al., 2012 AlO<sub>x</sub>-based Resistive Switching Device with Gradual Resistance Modulation for Neuromorphic Device Application in Proc. of IEEE International Memory Workshop.
262. Yi Li, et.al., 2014 Activity-Dependent Synaptic Plasticity of a Chalcogenide Electronic Synapse for Neuromorphic Systems Sci. Rep. 4, 4906, 1-7.
263. Valov I, Waser R, Jameson J R and Kozicki M N 2011 Electrochemical metallization memories—fundamentals, applications, prospects Nanotechnology 22, 254003.
264. Sakamoto T, Sunamura H, Kawaura H, Hasegawa T, Nakayama T and Aono M 2003 Nanometer-scale switches using copper sulfide Appl. Phys. Lett. 82, 3032–4.
265. Russo U, Kamalanathan D, Ielmini D, Lacaita A L and Kozicki M N 2009 Study of multilevel programming in programmable metallization cell (PMC) memory IEEE Trans. Electron Devices 56 1040–7
266. Jo S H, Chang T, Ebong I, Bhadviya B B, Mazumder P and Lu W 2010 Nanoscale memristor device as synapse in neuromorphic systems Nano Lett. 10 1297–301.
267. 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 Nature Mater. 10 591–5.
268. Nayak A, Ohno T, Tsuruoka T, Terabe K, Hasegawa T, Gimzewski J K and Aono M 2012 Controlling the synaptic plasticity of a Cu<sub>2</sub>S gap-type atomic switch Adv. Funct. Mater. 22 3606–13.
269. 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 IEDM Dig. Tech. Papers, 10.3.1–4.
270. Rawan Naous, Maruan Al Shedivat, Emre Neftci, Gert Cauwenberghs, and Khaled Nabil Salama, 2016 Memristor-based neural networks: Synaptic versus neuronal stochasticity, AIP Advances 6, 111304.
271. Yunus Babacan, Fırat Kaçar and Koray Gürkan, 2016 A spiking and bursting neuron circuit based on memristor, Neurocomputing 203, 86-91.
272. Angeliki Pantazi, Stanisław Woźniak, Tomas Tuma and Evangelos Eleftheriou, 2016 All-memristive neuromorphic computing with level tuned neurons, Nanotechnology 27 355205.
273. Tomas Tuma, Angeliki Pantazi, Manuel Le Gallo, Abu Sebastian and Evangelos Eleftheriou, 2016 Stochastic phase-change neurons, Nature Nanotechnology 11, 693–699
274. Alex M. Andrew, 2003 Spiking Neuron Models: Single Neurons, Populations, Plasticity, Kybernetes 32 (7/8).
275. Sander M.Bohte, Joost N.Kok and Han L Poutre, 2002 Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing 48(1–4), 17-37.
276. E.M. Izhikevich, 2004 Which model to use for cortical spiking neurons? IEEE Transactions on Neural Networks 15(5), 1063–1070.
277. Bruno A Olshausen and David J Field, 2004 Sparse coding of sensory inputs, Current Opinion in



- Neurobiology 14(4), 481-487.
278. Alexander Borst and Frédéric E. Theunissen, 1999 Information theory and neural coding, *Nature Neuroscience* 2, 947–957.
279. S. Burç Eryilmaz, et. al., 2013 Experimental Demonstration of Array-level Learning with Phase Change Synaptic Devices IEDM, *Dig. Tech. Papers*, 621-624.
280. Tomas Tuma, Manuel Le Gallo, Abu Sebastianand and Evangelos Eleftheriou, 2016 Detecting Correlations Using Phase-Change Neurons and Synapses, *IEEE Elec. Dev. Lett.* 37(9),1238-1241.
281. Elena Ioana and Lorena Anghel, 2017 Fully connected Single layer STT-MTJ based Spiking Neural Network under process Variability in *Proc. of IEEE/ACM Int. Sym. on Nanoscale Architectures (NANOARCH)*.
282. G. Pedretti, et. al., 2017 Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity *Sci. Rep.* 7: 5288,1-10.
283. D. Ielmini, 2018 Brain Inspired Computing with resistive switching memory(RRAM):Devices, Synapses and Networks *Microelectronic Engineering* 190, 44–53.
284. Thilo Werner, et.al., 2016 Spiking Neural Networks Based on OxRAM Synapses for Real-Time Unsupervised Spike Sorting *Front. Neurosci.* 10(474), 1-12.
285. Snaider Carrillo, et.al., 2013 Scalable Hierarchical Network-on-Chip Architecture for Spiking Neural Network Hardware Implementations, *IEEE Transactions on Parallel and Distributed Systems* 24(12), 2451 – 2461.
286. Dongjoo Shin, Jinmook Lee, Jinsu Lee and Hoi-Jun Yoo, 2017 DNPU: An 8.1TOPS/W Reconfigurable CNN-RNN Processor for General-Purpose Deep Neural Networks, *ISSCC Dig. Tech. Papers*, 240-242.

## Figures & Tables

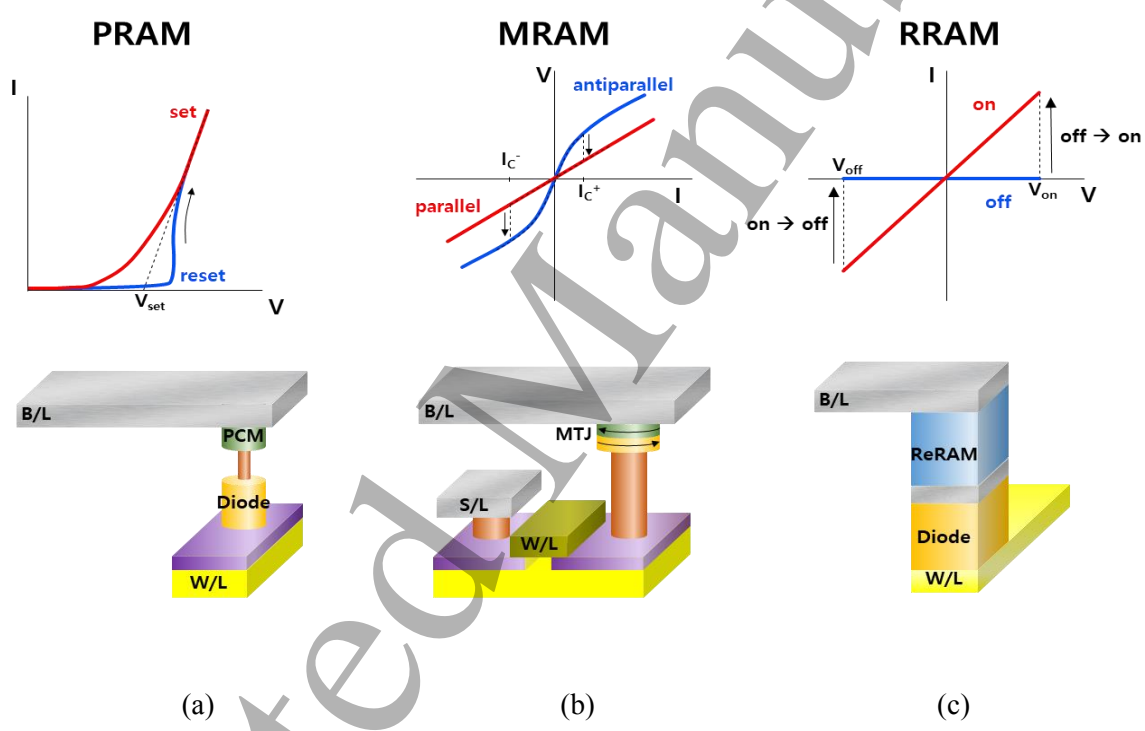
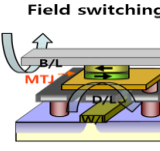
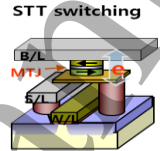
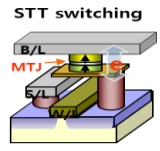


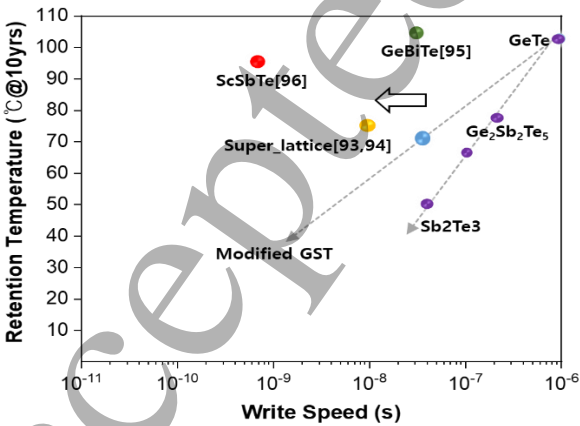
Fig. 1 The various cell structures and I-V characteristics of different types of memristors which show explicit pinch-off in I-V curve representing memristor characteristics. The cell structures are represented as (a) PRAM, (b) MRAM(STT-MRAM) and (c) RRAM.

|                                   | NAND   | STT-MRAM                     | PRAM  |   |                             | RRAM                      |                                     |
|-----------------------------------|--|------------------------------|---|---|-----------------------------|---------------------------|-------------------------------------|
| Status                            | Matured <sup>[75]</sup>                                      | Prototype <sup>[69,70]</sup> | Research                                      | Prototype <sup>[71,72]</sup>                | Research                    | Prototype <sup>[74]</sup> | Research                            |
| Feature size                      | 4F <sup>2</sup>  | ~ 46F <sup>2</sup>           | 6F <sup>2</sup>                               | 4F <sup>2</sup>                             | 4F <sup>2</sup>             | > 4F <sup>2</sup>         | 4F <sup>2</sup>                     |
| Maximum Density                   | 512Gb  | 8Mb, 256Mb <sup>[70]</sup>   | 4Gb <sup>[77]</sup>                           | 1Gb <sup>[72]</sup> -128Gb <sup>[71]</sup>  | 128Gb                       | 4Mb                       | 32Gb <sup>[101]</sup>               |
| Cell Area(nm <sup>2</sup> )       | < 600<br>( Vertical Cell )                                   | 36400                        | 7500 <sup>[82]</sup>                          | 800 <sup>[71]</sup> - 6084 <sup>[72]</sup>  | 2.3 <sup>[98]</sup>         | -                         | 3 <sup>[102]</sup>                  |
| MLC                               | 4bits  | 1bit                         | 1bit  | 1bit  | > 3bits <sup>[99]</sup>     | 1bit                      | > 3bits <sup>[103]</sup>            |
| Bit Access                        | Page: 4kB<br>Block: >1M                                      | Byte                         | Byte  | Byte  | Byte                        | Byte                      | Byte                                |
| Write voltage<br>( Cell Only )    | > 20V  | < 1.0V                       | 0.27V <sup>[86]</sup><br>2.4V <sup>[87]</sup> | < 2.0V                                      | < 1.5V <sup>[97]</sup>      | < 1.5V                    | < 1.0V <sup>[104]</sup>             |
| Write time                        | 80us /10ms   | 50ns                         | 0.2ns <sup>[82]</sup>                         | 500ns <sup>[72]</sup> ~30us <sup>[71]</sup> | < 1ns <sup>[96]</sup>       | 6400ns                    | 0.3– 230us <sup>[101,105-107]</sup> |
| Endurance                         | Single bit:10 <sup>5</sup><br>Two bits:<br>3x10 <sup>3</sup> | >10 <sup>12</sup>            | >10 <sup>12</sup> [84,85]                     | 10 <sup>9</sup>                             | > 1X10 <sup>12</sup> [67]   | 1.2X10 <sup>6</sup>       | 10 <sup>12</sup> [108]              |
| Retention time<br>(extrapolation) | >10 yr@85 ℃  | >10 yr@85 ℃                  | >10 yr@85 ℃                                   | 10 yr@70 ℃                                  | 10yr@220 ℃ <sup>[100]</sup> | >10yr@85 ℃                | 10yr@150 ℃ <sup>[109]</sup>         |
| Write energy/bit                  | > 160pJ  | ~1pJ                         | 6.24X10fJ <sup>[88]</sup>                     | 30pJ <sup>[72]</sup>                        | ~0.08pJ <sup>[98]</sup>     | -                         | 0.1pJ~10nJ <sup>[110]</sup>         |
| Logic Compatibility               | Very Poor  | Good                         |   |   |                             |                           |                                     |

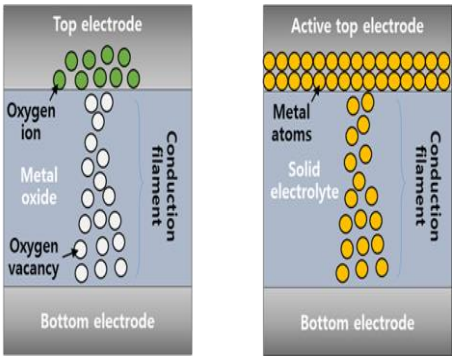
Tab.1 The comparison table represents the characteristics between memristor devices referring to NAND Flash memory. This new type of comparison table show the characteristics in research and prototype level, which represent the possibility and matured data respectively.

| Types               |                   | Conventional MRAM  | STT-MRAM ( Planar MTJ )   | STT-MRAM ( p-MTJ )   |
|---------------------|-------------------|--|---|--|
| Schematic View      |                   |  |  |  |
| Ideal Cell Size     |                   | $> 22.5F^2$  | $8F^2$  | $6F^2$   |
| Scalability         | Technology Node   | $\sim 90\text{nm}$   | $20\text{nm} \sim 40\text{nm}$  | $8\sim 20\text{nm}$ @ Interface<br>$5 \sim 10\text{nm}$ @ Bulk                       |
|                     | Switching Current | MTJ size $\downarrow \rightarrow I_{sw} \uparrow$                                  | MTJ size $\downarrow \rightarrow I_{sw} \downarrow$                                 | MTJ size $\downarrow \rightarrow I_{sw} \downarrow$                                  |
|                     | MTJ A/R           | $> 2:1$  | $> 2:1$   | $\sim 1:1$   |
| An-isotropy(erg/cc) |                   | Shape, $K_u \sim 10^4$   | Shape, $K_u \sim 10^4$  | Interface: $K_u \sim 5 \times 10^6$<br>Bulk : $K_u \sim 10^7$                        |

(a)



(b)



OxRAM

CBRAM

(c)

Fig. 2 The key attributes for memristor devices. (a) The evolution of MTJ device technology for higher density and more scalability of MRAM. (b) The development trend of phase change materials to achieve fast write speed for energy efficient PRAM. (c) The representative operation mechanisms for RRAM devices: oxide RAM(OxRAM) and conductive bridge RAM (CBRAM).

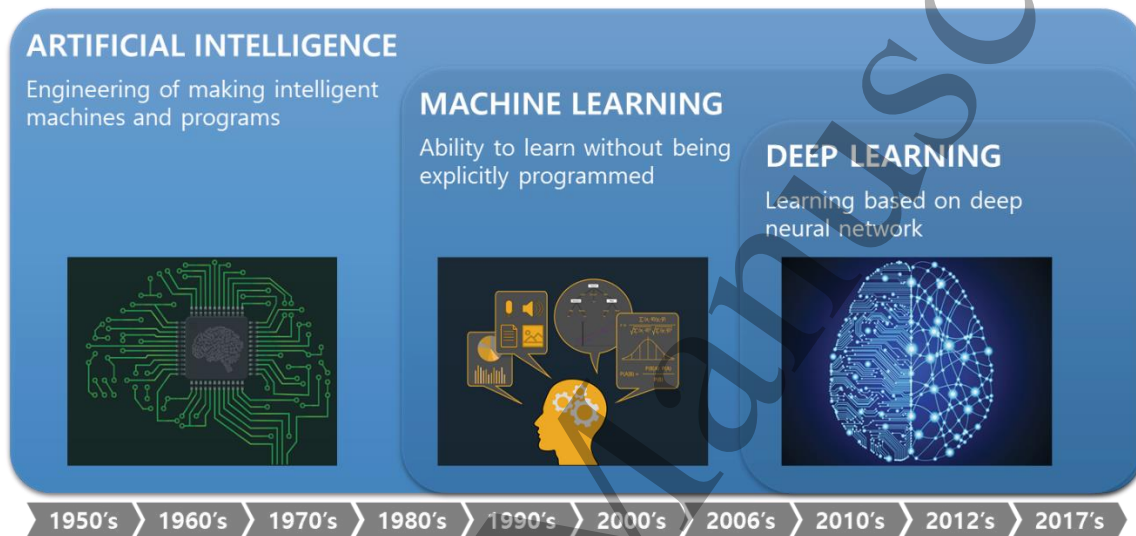


Fig. 3 The relationship between A.I., machine learning, and deep learning. Even though, there is a still debate, this concept has been generally accepted, which is mainly cited from NVIDIA, <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai>.

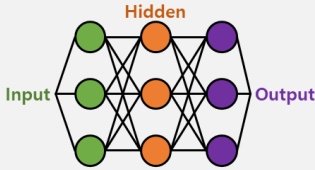
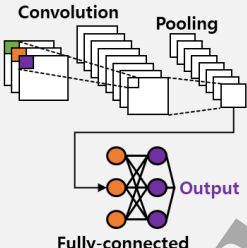
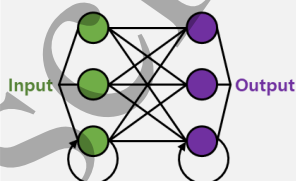
|                   | MLP<br>(multi-layer perceptron)   | CNN<br>(convolution neural network)  | RNN<br>(recurrent neural network)   |
|-------------------|---|--|---|
|                   |  |  |  |
| Characteristic    | Fully-connected Layer   | Convolution layer  | Feedback path, internal state   |
| Major Application | Simple classification<br>Hand-written letter recognition, ...                     | Vision processing,<br>Face recognition, image classification, ...                  | Sequential data processing, transition, speech recognition, ...                     |
| Layers            | 3 – 10 layers   | 5 – 100 layers   | 3 -5 layers   |

Fig.4 The backbone architectures of neural networks: Multi-layer Perceptron (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN)[286].

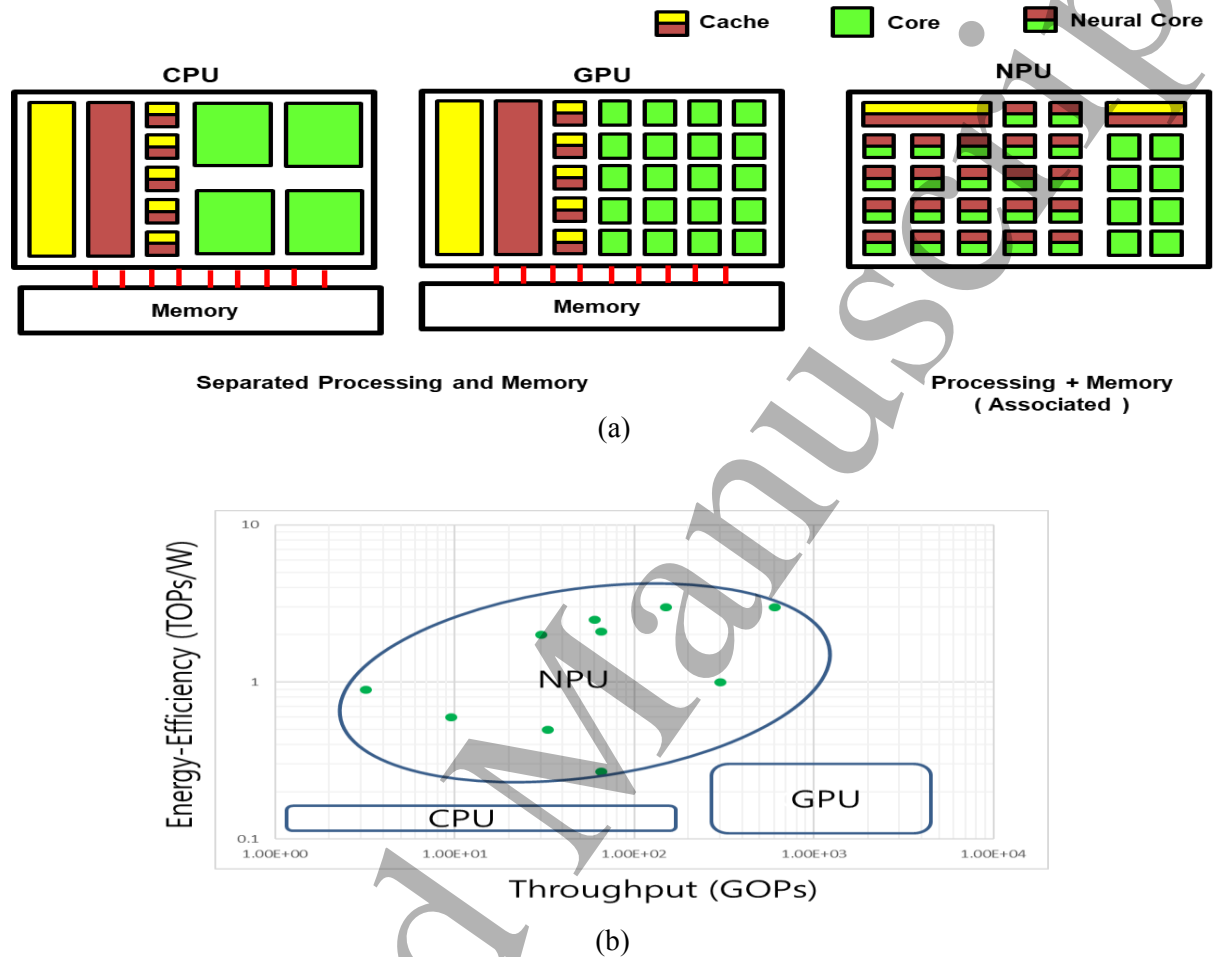


Fig.5 (a) The main differences between central processing units(CPUs), graphics processing units (GPUs) are the number and the organization of cores. The big difference between conventional processing units(CPUs and GPUs) and NPUs is separated or associated processing and memory in operation and organization. The NPUs operate processing and memory simultaneously during the computation which is very similar to the brain operation. (b) An overview of the reported performance of the deep neural network processors published at the International Solid-State Circuits Conference in 2016 and 2017. The energy-efficiency of NPUs are much better than conventional CPUs and GPUs.

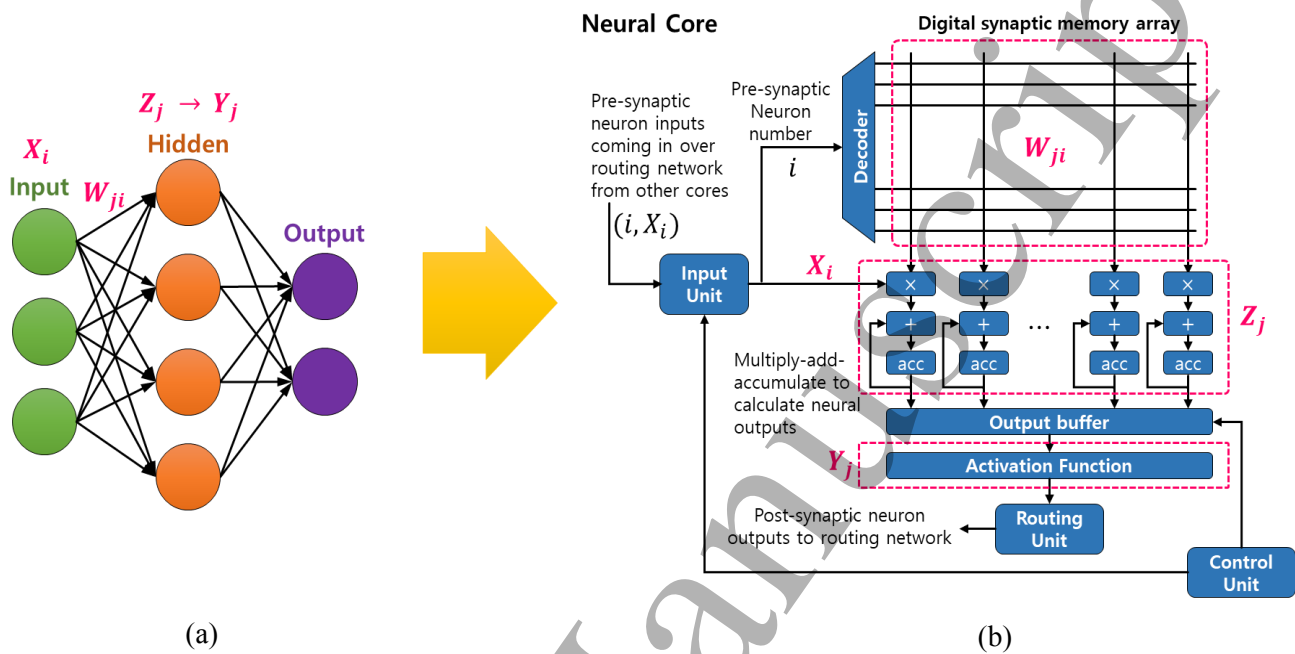


Fig.6 The idea of neural network unit can be implemented in circuit design for the core of NPUs is shown in figure. The basic unit to constitute the neural networks as shown in (a). This neural network can be realized through semiconductor technologies as shown in (b). The function of input/output layers (neurons) including multiplication, add and activation can be established by CMOS digital or analog logic circuit and weight storage function (synapses) can be established by memory devices such as SRAM, DRAM and memristors [189].



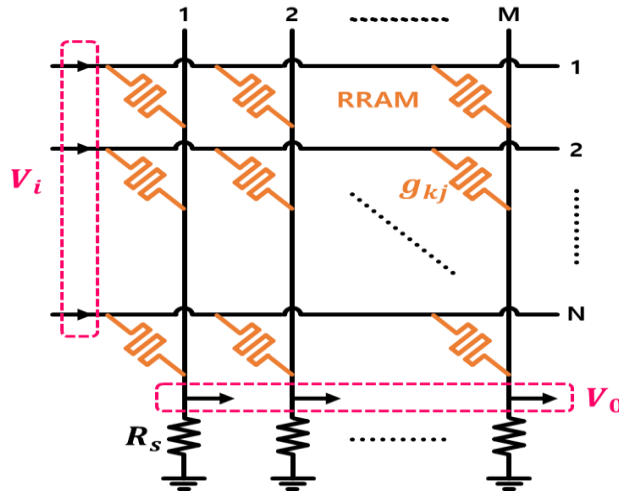
| Neural Chips                | Technology Node | Neuron   | Synapses | Remark  |
|-----------------------------|-----------------|----------|----------|---|
| True North <sup>[38]</sup>  | 28 nm           | CMOS     | SRAM     | The main role of synapses is to store weights like memory usage |
| Spinnaker <sup>[36]</sup>   | 130 nm          | ARM Core | DRAM     |   |
| Neurogrid <sup>[190]</sup>  | 180 nm          | Analog   | SRAM     |   |
| Da_DianNao <sup>[191]</sup> | 28 nm           | CMOS     | e-DRAM   |   |

Tab.2 Recently developed neural processors including accelerators have been displayed in this table.

Even though their approaches are a little different in technology node and neuron devices, but their synapses have been achieved by volatile memory devices such as SRAM and DRAM to store weights.

| Application                      | Conventional         | High Performance     | High Density                    | Remark                       |
|----------------------------------|----------------------|----------------------|---------------------------------|------------------------------|
| Synapse                          | SRAM <sup>[38]</sup> | MRAM <sup>[69]</sup> | PRAM <sup>[71]</sup>            |                              |
| Density                          | 256M                 | > 1Gb                | > 128Gb                         |                              |
| Cell Size<br>( $\mu\text{m}^2$ ) | 0.15~0.2@28nm        | 0.036@28nm           | 0.0004@20nm                     |                              |
| Latency                          | < 10ns               | 20ns ~30ns           | 500ns (Write)<br>< 100ns (Read) |                              |
| Endurance                        | $10^{15}$            | $10^{15}$            | $10^9$                          | On-line learning<br>> $10^9$ |
| Nonvolatility                    | No                   | Yes                  | Yes                             |                              |

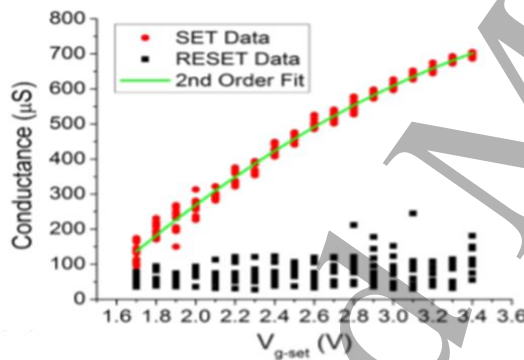
Tab. 3 The comparison table of characteristics for synaptic devices according to applying different synaptic device technologies such as SRAM, MRAM and PRAM which are currently available technologies in real neural network applications.



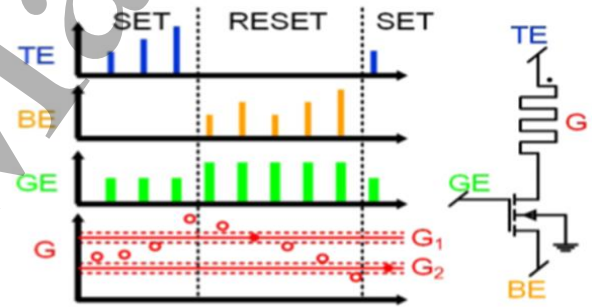
$$V_{ij} = r \cdot \sum_k (V_{ik} \cdot g_{kj}), \quad g_{kj} = \frac{1}{M_{kj}}$$

1 Memristor = 1 Adder + 1 Multiplier + 1 SRAM

(a)



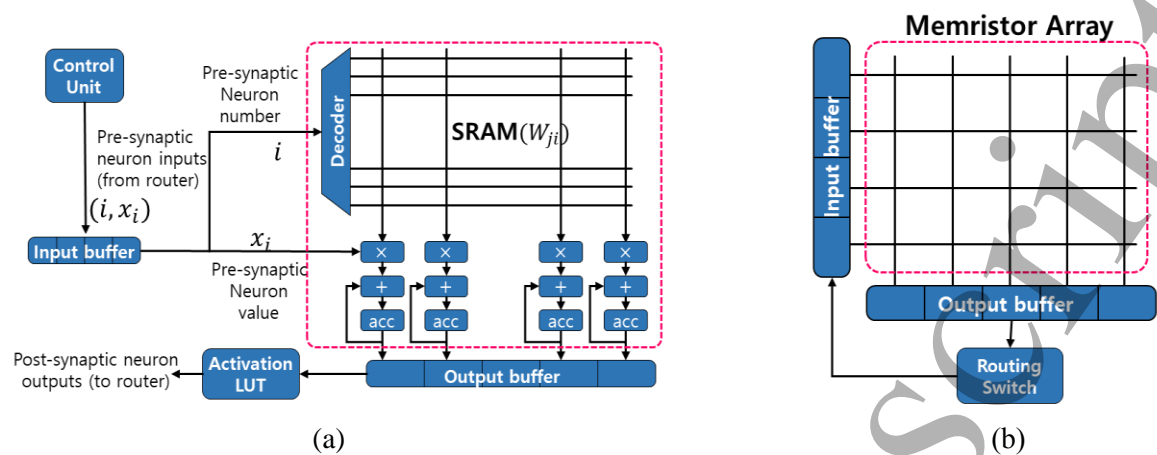
(b)



(c)

Fig. 7. The memristor crossbar array can naturally transfer the weighted combination of input signals to output voltages and realize the matrix-vector multiplication. The continuous variable resistance states of memristor devices enable a wide range of weight matrices that can be represented by the crossbar shown in (a) [199]. The conductance values as a function of set voltage following SET-RESET switching in a TaOx based memrsitor device for matrix-vector multiplication shown in Fig. 7(b). Fig. 7(c) shows a representation of the procedure for programming the memristor device [202]. Reprinted with permission from [202] Copyright

2016 IOP Publishing.



| Deep Neural Network |                        |           |                            | Edge Detection    |                        |           |                            |
|---------------------|------------------------|-----------|----------------------------|-------------------|------------------------|-----------|----------------------------|
|                     | Area(mm <sup>2</sup> ) | Power(mW) | Efficiency<br>over<br>RISC |                   | Area(mm <sup>2</sup> ) | Power(mW) | Efficiency<br>over<br>RISC |
| RISC                | 472.65                 | 78474.0   | 1                          | RISC              | 125.76                 | 20880.0   | 1                          |
| NN SRAM             | 1.88                   | 82.4      | 952                        | NN SRAM           | 3.75                   | 433.16    | 48                         |
| NN<br>Mem.(Store)   | 0.64                   | <10       | > 7,847                    | NN<br>Mem.(Store) | 1.04                   | <20       | >1044                      |
| NN<br>Mem.(MAC)     | 0.08                   | 0.42      | 187,064                    | NN<br>Mem.(MAC)   | 0.13                   | 1.41      | 14,813                     |

(c)

Fig. 8 The schematic diagrams for (a)weight storage and (b)matrix-vector multiplication operation are shown. Matrix-vector multiplication operation does not need multiplication and accumulation logic circuit outside synapse(memristor) array due to the unique characteristics of memristor. The comparison of simulation results for different types of neural networks based on weight storage and matrix-vector multiplication operation technology is shown. For this comparison the processing of real time application was examined: For the deep network, inputs having 28×28 pixel handwritten digits

was used with processing 100,000 characters/sec. For edge detection 1280×1080 image stream at 60 frames per second was processed [205].

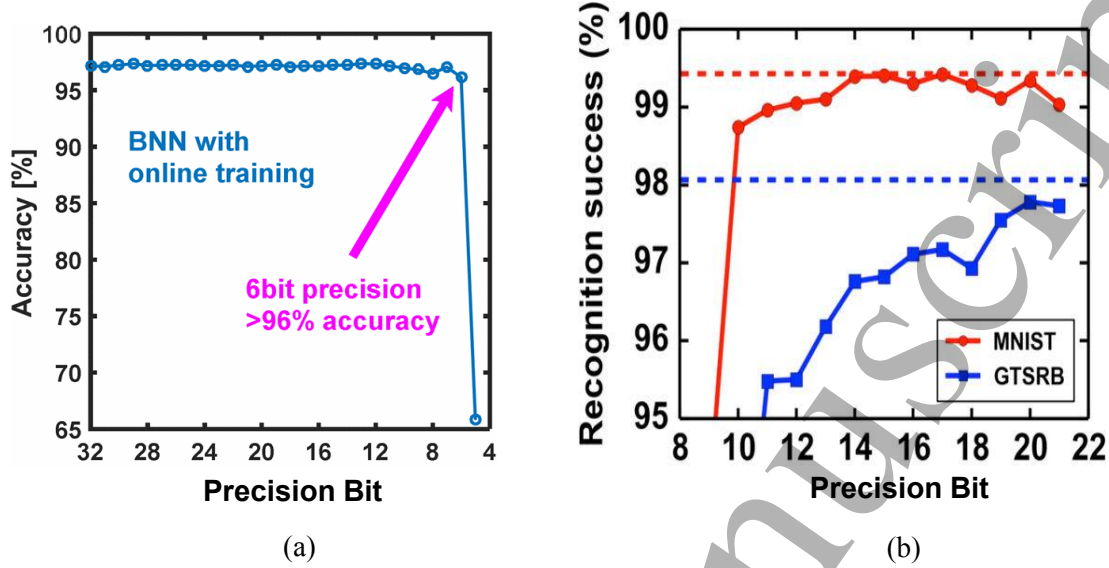


Fig. 9. The accuracy vs. resolution of weights/neurons in some different cases. At least 6 bit-resolution is required for MNIST classification in MLP[210], but the requirement of bit resolution for MNIST in CNN increases up to 10 bit. Moreover more complicated application such as traffic sign classification(GTSRB database) requires much more resolution of weights[211]. Reprinted with permission from [210] Copyright 2016 IEEE and [211] Copyright 2015 IEEE.

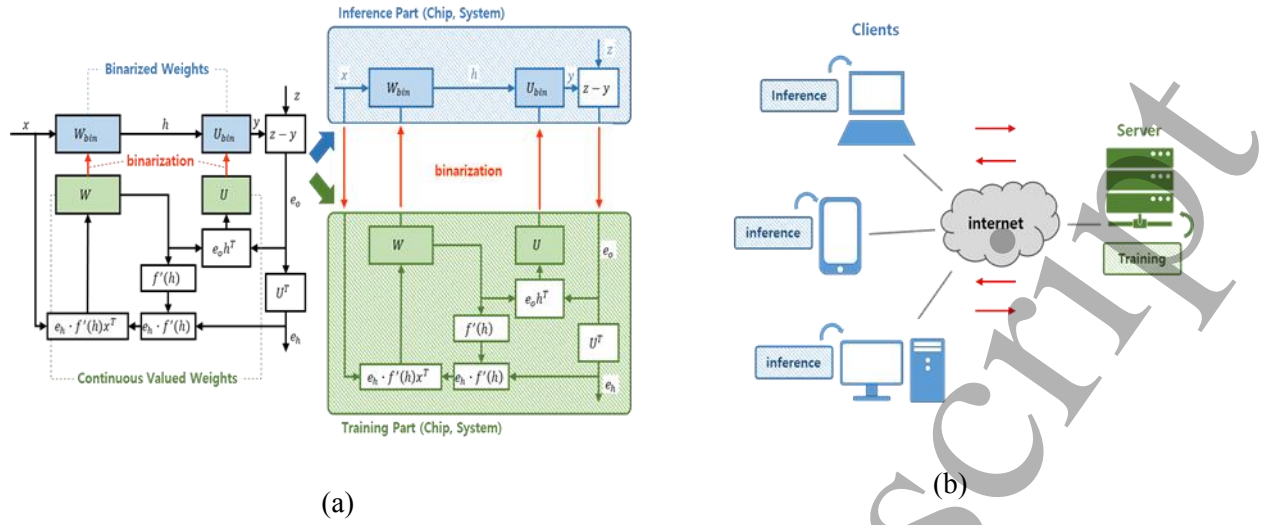
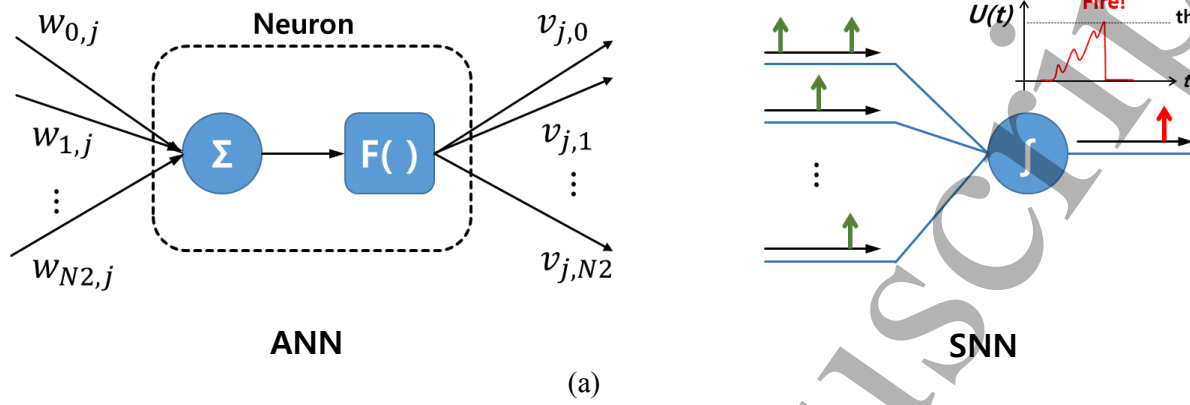


Fig. 10. The neural hardware can be divided into inference and training units, as shown in Fig. 10(a).

Unlike training, inference does not require high precision bits of weight. Therefore, BNN is adoptable for inference machine which lead to drastically reduced memory size and computing energy. Fig.10(b) shows the schematic of operation methodology for separated inference and training system. Client devices like mobile phone and tablet PC having not enough memory space for deep neural network applications are very well matched to inference machine. This machine can get trained results from servers which have enough size. We can operate very efficiently the computing system by utilizing each device's and neural network's characteristics.



| Application                        | ANN                         | SNN                                | Remark  |
|------------------------------------|-----------------------------|------------------------------------|---|
| Synapses Devices                   | Digital, Analog Memory      | Analog Circuit, Memory             | SNN requires the characteristics of Analog memristor                |
| Neurons                            | Activation Functions        | (Leaky) Integration & Fire, etc.   | The mimicking LIF, STDP requires large area and complicated circuit |
| Training                           | Back-propagation            | STDP, Hebb's law, Back-propagation | More practical approaches have been studied for SNN                 |
| Neuronal Activations               | Multi-level value           | Timing Domain Coded Spikes         | -   |
| Represent negative neuronal values | Negative Activation Value   | Inhibitory Neurons                 | -   |
| Chip Operation                     | Synchronized on Clock Cycle | Event Driven                       | The faster latency is required due to event-driven operation in SNN |
| Theoretical sources                | Mathematical derivation     | Brain Enlightenment                | Better Energy Efficiency in SNN                                     |

(b)

Fig. 11. The comparison between ANN and SNN is described in operation scheme(a) and table(b).

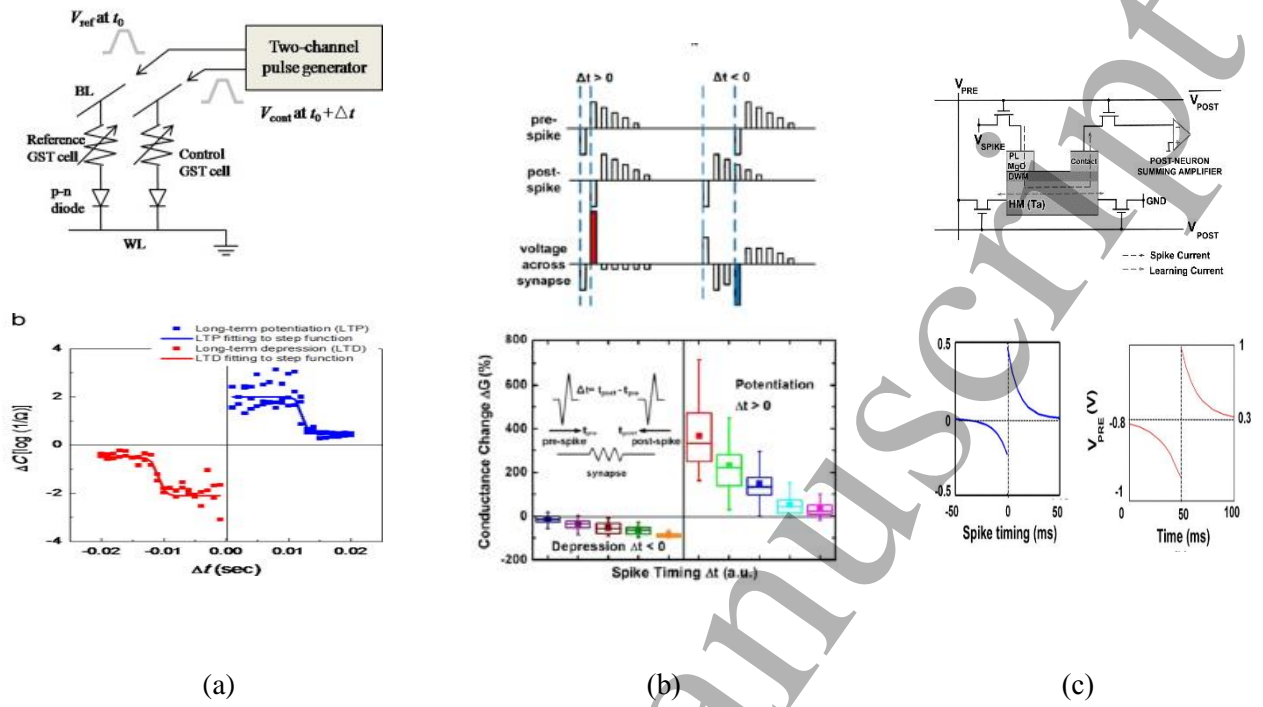


Fig.12 The Spike Timing Dependent Plasticity (STDP) function for SNN can be easily emulated by using memristor devices such as PRAM, RRAM and MRAM. Fig. 12(a) represents the circuit diagram of 2-PRAM cells and emulated result of STDP function [17]. Fig. 12(b) shows that STDP has been implemented in HfOx RRAM synaptic devices using different pulse schemes which have gradual programming of resistive synaptic devices [256]. Fig.12(c) shows the MRAM based synaptic device with access transistors for separate spike transmission and learning current paths and its STDP characteristics [248]. Reprinted with permission from [17] Copyright 2015 Elsevier B.V. , [256] Copyright 2011 IEEE and [248] Copyright 2015 AIP.



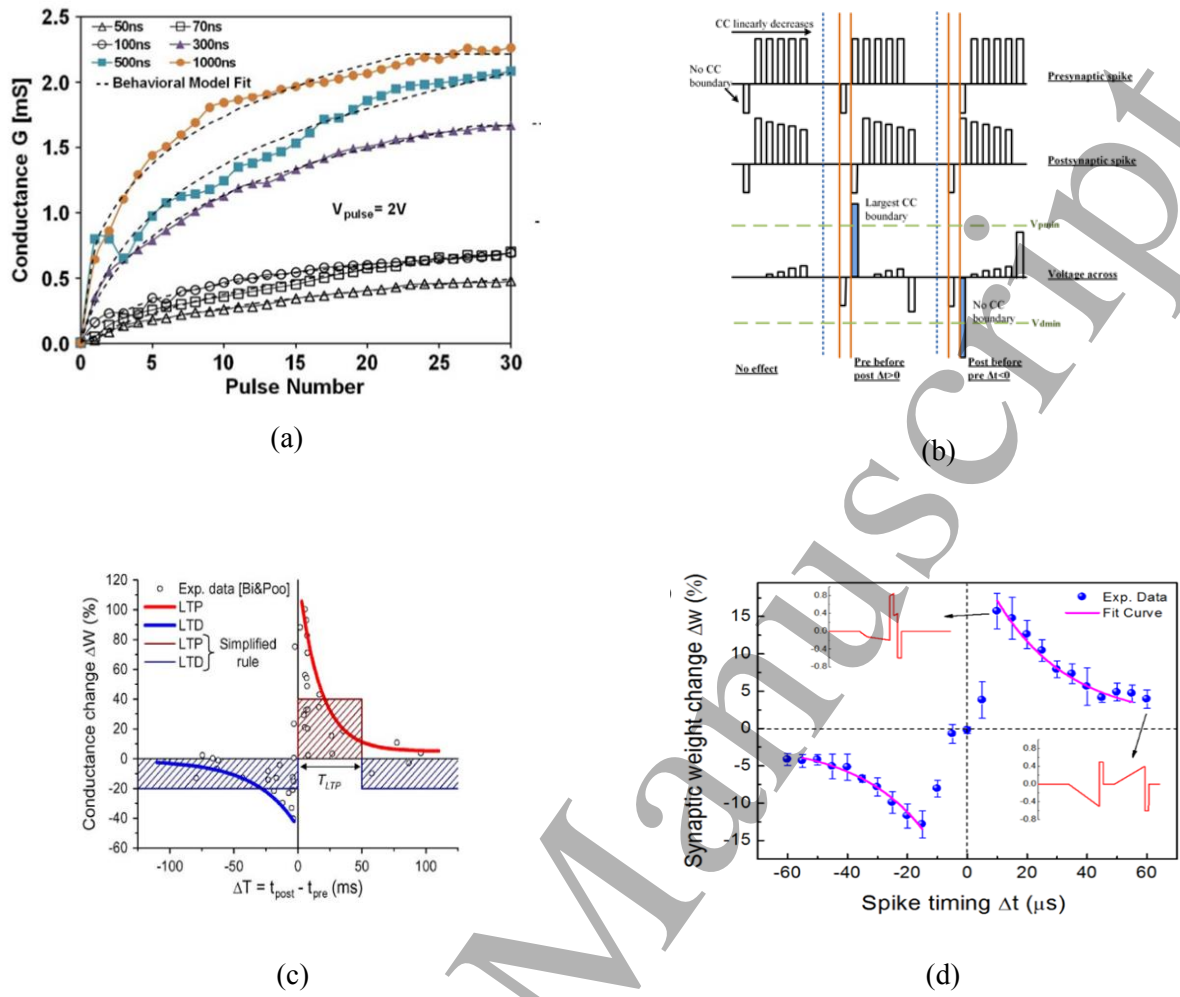


Fig.13 The conductance modification by the sequential number of an identical pulse and by the increasing voltage of the pulse, as shown in Fig. 13(a)[247] and 13(b) [200], respectively. Fig. 13(c)[247] shows the executed simplified STDP learning rule by the modifying synaptic by the identical pulse train. The result of STDP learning rule by applying different voltage effectively is shown in Fig. 13(d)[262]. Reprinted with permission from [247] Copyright 2012 IEEE and [262] Copyright 2014 Nature Publishing Group.

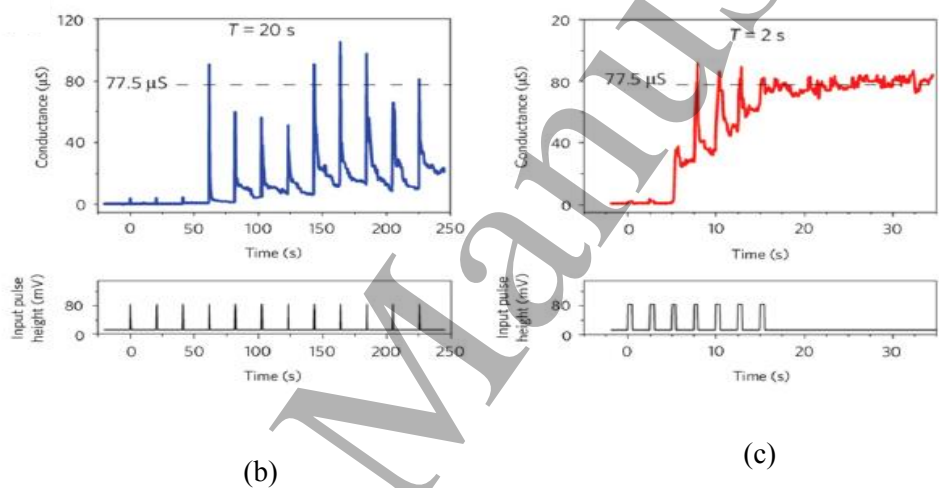
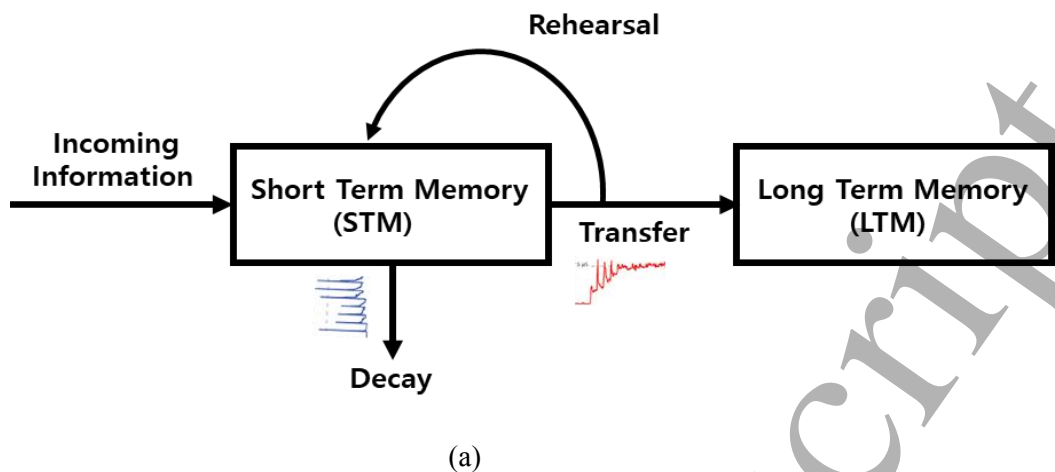


Fig. 14 A schematic diagram of biological memory architecture is drawn in (a).

Ag/Ag<sub>2</sub>S/nanogap/Pt synaptic device when input pulses( $V=80mV$ ,  $W=0.5$  s) were applied with intervals of (b) 20 s and (c) 2 s. Reprinted with permission from [267] Copyright 2011 Nature Publishing Group.

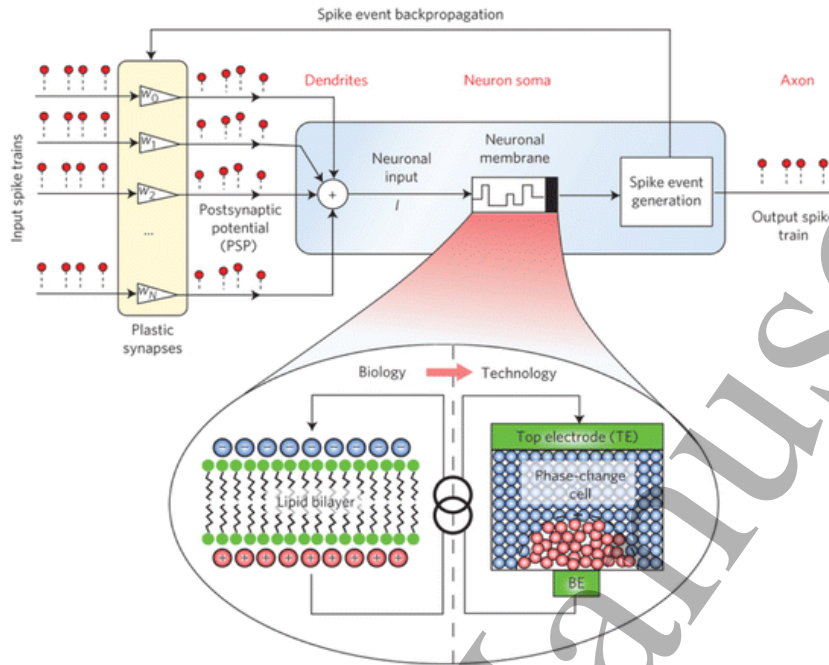


Fig. 15 The schematic diagram for emulating I&F neuron by using PRAM operation principle.

Reprinted with permission from [273] Copyright 2016 Nature Publishing Group

| Synapse Device            | Training        | Neuron Device | Neuron Function | Number of Synapse        | Application                       | Remark                              |
|---------------------------|-----------------|---------------|-----------------|--------------------------|-----------------------------------|-------------------------------------|
| PRAM <sup>[241]</sup>     | Simplified STDP | CMOS          | LIF             | 16,384(128x128)x60+60x10 | Traffic Detection                 | 1synapse=2PRAM                      |
| PRAM <sup>[274]</sup>     | Hebb's law      | CMOS          | I&F             | 10x10                    | Clustering of simple two patterns | 1synapse=1PRAM                      |
| PRAM <sup>[275]</sup>     | STDP            | PRAM          | I&F             | 10                       | Demonstrate Basic Concept         | Fully Memristor based Neural Device |
| RRAM <sup>[279]</sup>     | STDP            | CMOS          | LIF             | 32x5                     | Detection of Bio signal           | -                                   |
| RRAM <sup>[277,278]</sup> | STDP            | CMOS          | I&F             | 9(3x3)x2                 | Clustering of simple two patterns | 1synapse=1RRAM                      |
| RRAM <sup>[194]</sup>     | SRDP            | CMOS          | I&F             | 784×500+500×10           | Classification of MNIST           | Transferring ANN to SNN             |
| MRAM <sup>[276]</sup>     | Simplified STDP | CMOS          | I&F             | Simulation               | MNIST                             | 1synapse= 1~16 MTJs                 |

Tab. 4 The comparison table of various type SNNs by using different memristor devices.