SCIENCE CHINA
Information Sciences

• REVIEW •

June 2018, Vol. 61 060423:1–060423:14
https://doi.org/10.1007/s11432-018-9425-1

Special Focus on Memristive Devices and Neuromorphic Computing

Artificial neural networks based on memristive devices

Vignesh RAVICHANDRAN, Can LI, Ali BANAGOZAR,
J. Joshua YANG & Qiangfei XIA*

Department of Electrical and Computer Engineering, University of Massachusetts, Amherst MA 01003*, USA*

Received 22 February 2018/Accepted 27 March 2018/Published online 15 May 2018

Abstract The advent of memristive devices and the continuing research and development in the field of neuromorphic computing show great potential as an alternative to traditional von Neumann computing and bring us ever closer to realizing a true "thinking machine". Novel neural network architectures and algorithms inspired by the brain are becoming more and more attractive to keep up with computing needs, relying on intrinsic parallelism and reduced power consumption to outperform more conventional computing methods. This article provides an overview of various neural networks with an emphasis on networks based on memristive emerging devices, with the advantages of memristor neural networks compared with pure complementary metal oxide semiconductor (CMOS) implementations. A general description of neural networks is presented, followed by a survey of prominent CMOS networks, and finally networks implemented using emerging memristive devices are discussed, along with the motivation for developing memristor based networks and the computational potential these networks possess.

Keywords neuromorphic computing, memristors, neural networks

Citation Ravichandran V, Li C, Banagozar A, et al. Artificial neural networks based on memristive devices. Sci China Inf Sci, 2018, 61(6): 060423, https://doi.org/10.1007/s11432-018-9425-1

1 Introduction

The demand for cheaper and more powerful computers has only increased with time, which was possible by scaling down the size of complementary metal oxide semiconductor (CMOS) devices, thus allowing for greater device densities and performance as well as improved circuit designs on chips. While Moore's law [1] has been a good indicator of on-chip device density for the last several decades, the era of Moore's law is reaching an end as silicon based devices reach a physical limitation to their miniaturization [2], which serves as a major contributing factor in the quest to find suitable alternatives. Although the digital computer was touted as the universal machine for the last several decades, the traditional architecture, as suggested by Neumann [3], relies on a complex central processing unit (CPU) that sequentially accesses a memory bank to perform instructions in a serial manner [4]. While this setup has been shown useful in solving a multitude of different problems, it introduces an intrinsic limit on throughput by creating a major bottleneck in modern computers due to latency from the time required to access data in memory and transfer it to the processing unit to actually decode and perform instructions. This frequent data shuffling process also incurs huge energy consumption. Though processors have gotten faster and faster over time, they are limited by this data transfer rate, and the modern trend of Big Data and the Internet

* Corresponding author (email: qxia@umass.edu)

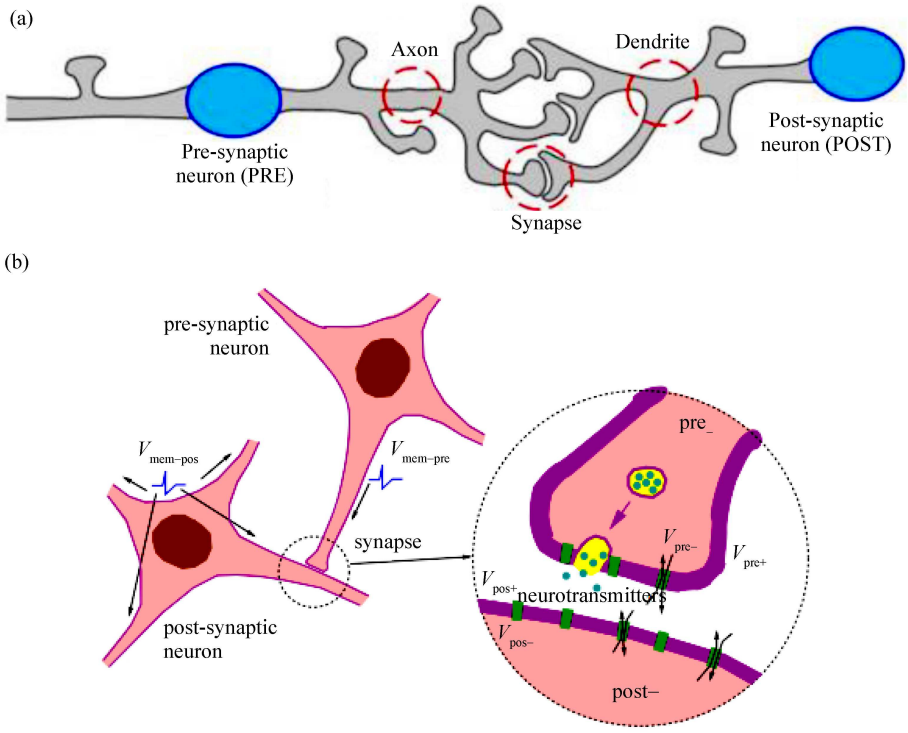

Figure 1 (Color online) Neuron and synapse structure. (a) Simplified biological structure of pre-synaptic and post-synaptic neurons showing the axon, synapses, and dendrites [6] @Copyright 2017 Scientific Reports; (b) biological synaptic structure and function [7] @Copyright 2011 Frontiers in Neuroscience.

of things (IoT) has made it necessary to find novel ways to bypass the von Neumann bottleneck, as well as the physical constraints placed upon the devices that the current architecture is composed of to allow for more timely and efficient processing.

The aforementioned issues with modern CMOS computing have been major motivating factors to come up with better solutions to today's computing problems. Though still in its infancy, neuromorphic computing is a computing paradigm inspired by neurobiology, aiming to emulate various anatomical and physiological features of the brain and nervous system to achieve more powerful and energy efficient computation. Drawing from a wide variety of disciplines such as biology, engineering, physics, and computer science, neuromorphic computing may be the last frontier of computing. Unlike the von Neumann architecture, which is limited by its serial nature, neuromorphic computing emphasizes massive parallelism and utilizes interconnected, simple processing units operating asynchronously in a manner similar to that of the biological brain [4]. The brain is composed of a complex network of cells, referred to as neurons, that propagate signals through their respective axons to communicate with other neurons [5]. Once the neuron has reached an intrinsic threshold, these axons carry action potentials which then elicit the propagation of various electrical and chemical signals to neighboring neurons' dendrites via synapses, as shown in Figure 1. The neurons individually serve as the brain's simple processing unit, using incoming stimuli from neighboring cells to decide whether or not to fire a signal to other neurons. The culmination of this complex network is what allows for many complex biological functions to arise including fine motor control, memory, and thoughts.

2 Neural networks

Neural networks (NNs) take inspiration from the biological brain and nervous system and use a complex system of connected artificial "neurons" to perform various computational tasks. These neurons are interconnected with each other via "synapses", which can be assigned a weight to modulate the strength

of an input being received by a neuron [4]. As mentioned previously, neural networks are composed of a network of neurons, also referred to as nodes, that are interconnected amongst each other to allow for the incoming stimulus to be processed and a corresponding output to be generated. To achieve this, such networks are generally organized into layers including an input layer, intermediate hidden layers, and an output layer, where each layer is composed of a number of nodes to allow for data processing.

Generally speaking, there are two main phases for NNs: The first one being the learning or training phase in order for the system to recognize specific incoming stimuli during the second phase—the inference phase. During the training phase, the weights and biases of the network are modified to allow it to accurately and efficiently perform a desired task. Training can be done either offline or online. Generally, offline training involves the network being trained before it can be used for the application it was designed for from a predetermined dataset. Online training, on the other hand, implies that the network is able to learn on the fly whilst in operation, staying true to the operation of biological counterparts [6, 8]. Along with online and offline training, there exist supervised and unsupervised training. Supervised training generally involves training a network with a predetermined dataset to allow the network to learn and develop an algorithm that provides an accurate output given a specific task. This type of training is used frequently in modern machine learning and is demonstrated by the network implemented by Li et al. [9], where the group tuned the conductance of the cells to map specific functions prior to using the network for a variety of tasks. On the other hand, unsupervised learning resembles what is found in biology, where the network is allowed to change in response to input stimulation in real time to arrive at an output. This type of learning is demonstrated in the network by Pedretti et al. [6] which utilizes spiking artificial "neurons" and plastic connections between them to learn and recognize static patterns, allowing it to more faithfully mimic the biological nervous system with similar learning rules. Learning in biology is usually associated with synaptic plasticity, with several proposed mechanisms on how various forms of plasticity manifest, including rate and time dependent plasticity. There are many learning rules that influence plasticity, one of the more popular being the Hebbian learning rule [10], which is commonly used in neural network applications. According to the Hebbian rule, the synaptic weights are strengthened if a post-synaptic neuron fires after it receives consistent stimulus from a pre-synaptic neuron, and vice versa if the post-synaptic cell consistently fires before it receives stimulus from a pre-synaptic neuron. By using such learning rules, neural networks are able to constantly adapt and improve their performance on their assigned tasks.

Prominent types of artificial neural networks (ANN) are discussed in the implementation section of this paper and various convolutional neural networks (CNN) and spiking neural networks (SNN) that have been simulated and experimentally demonstrated to complete specific tasks are reported. While level-based networks, such as CNNs, currently stand out for modern machine learning applications, SNNs show more potential for bioinspired computing since they more closely mimic biology. Although it is too soon to say which type of network will dominate in the future, neural networks as a whole can be used for a very broad range of computing applications including mimicking bioinspired plasticity, computer vision, pattern/object recognition, speech recognition, and efficient neuroscience modeling [6, 7, 9, 11–28]. Many of these applications are starting to become more and more prominent in today's world of Big Data [8].

3 CMOS based neural networks

Due to the advantages of neural networks and neuromorphic computing, several modern state-of-the art NNs using standard CMOS devices have been devised. In 2014, DaDianNao [29] was introduced, focusing on improving the performance of CNNs with a solution based on a custom multi-chip machine learning architecture that allowed for high-internal bandwidth, enabling a high degree of parallelism without significantly impacting area efficiency. After being tested on the largest neural networks at the time, the chip was claimed to be able to achieve a speedup of over 450 times over the standard graphics processing unit (GPU) implementation while using roughly 150 times less power when compared to the average power consumption of 64-chip systems. In the same year, IBM TrueNorth [30] was unveiled,

using 5.4 billion transistors to allow for one million spiking neurons and up to 256 million configurable synapses. The brain-inspired architecture aimed to provide a non-von Neumann solution and is suitable for many NN tasks. When tested in real time with a video of 400 pixels by 240 pixels at 30 fps, the chip consumed merely 63 mW.

In an effort to improve cost-energy performance and accelerate the inference phase of NNs, a custom application-specific integrated circuit (ASIC), the Google tensor processing unit (TPU) [31], was created. The unit was designed to work with pre-existing server hardware and served as a co-processor on the peripheral component interface (PCI) input/output (I/O) bus. It receives instructions directly from the host server with the ability to run entire inference models in an attempt to reduce interactions with the host CPU, allowing for reduced latencies. To carry out computations, the chip is equipped with a Matrix Multiply Unit that is capable of reading and writing 256 values per clock cycle and can perform matrix multiplication or convolution. When compared with server-class CPUs and GPUs for the same NN applications, the TPU has demonstrated to be between 15 and 30 times faster, while consuming less power. Along with novel TPU designs, GPU based solutions are still very common due to their intrinsic parallelism and numerous processing cores. The NVIDIA DGX [32] is one such system that attempts to accelerate deep learning.

Although the aforementioned approaches do have their benefits while used in NNs, it would not be practical to scale them to the complexity of the nervous system as they still cannot rival the power efficiency, compactness, or immense parallelism found in biology. This has driven the need to find alternatives to using CMOS technology in an attempt to find devices that behave in a manner similar to the brain at a more fundamental level.

4 Memristor based neuromorphic architecture

Proposed by Chua [33] in 1971, the memristor is the fourth circuit element along with the resistor, capacitor, and inductor. In 2008, under Williams, HP Labs developed a stable, solid-state device that was linked to the memristor [34–37]. As research continued, many useful traits of memristive devices were established [38–43]. Firstly, unlike modern dynamic random access memory (DRAM), which requires consistent power to store data, many types of memristors are able to retain their conductance states without the need for a power source. In addition to that, the two terminal nature of the device also allows for deeper scaling and increased area efficiency [6]. Memristors have also been shown to switch faster than standard DRAM [44] and although static random access memory (SRAM) may be able to compare performance wise, it is more expensive and still suffers from volatility and the scaling limits of CMOS technology. Thus, with proper implementations, memristive devices have the potential to reduce power consumption, improve switching speed, and increase device density compared to more traditional CMOS designs [20, 45–48].

Although some of the features and functions of neurons and synapses can be emulated using CMOS technology, the amount of power consumed, devices required, and on-chip area used make it impractical. On the other hand, using memristive devices over CMOS devices has many advantages including increased throughput due to parallelism, decreased power consumption, and increased device scalability and density on chip, as well as intrinsic analog properties [49]. Although different types of networks place different types of demands on the memristive devices used, memristors are still the fundamental building block of the networks, and thus the networks are able to reap the benefits of these advantages. While many level-based networks may rely heavily on accurate conduction tuning, other networks, such as SNNs may rely on more complex memristor dynamics and threshold switching to function. In light of this, recent studies have shown that certain types of memristive devices demonstrate behavior similar to synapses in the brain [11, 48, 50] and are able to use biological rules, such as spiking-time-dependent-plasticity (STDP), to update synaptic weights in a manner resembling biology [7, 21]. These attributes of memristors will no doubt prove to be useful as neuromorphic computing continues to develop.

Going from a device perspective to a network perspective, memristors have already shown a lot of

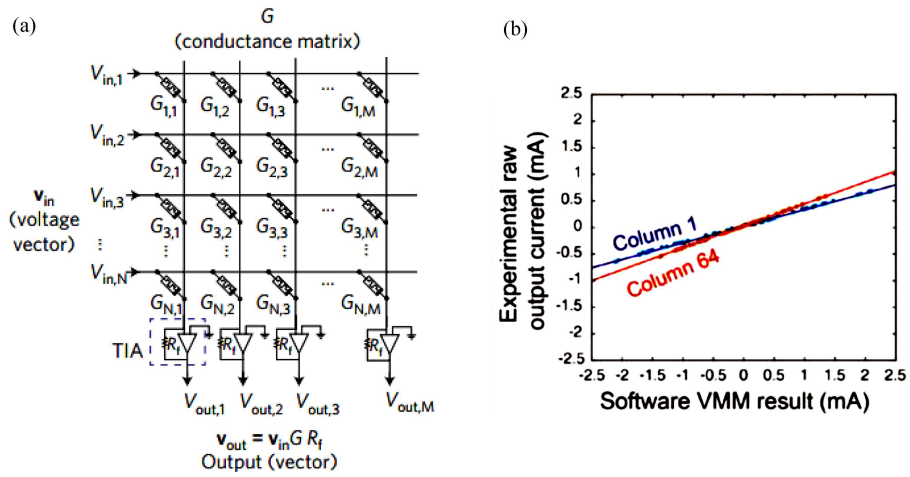

Figure 2 (Color online) (a) Memristor crossbar for vector matrix multiplication (VMM); (b) current (mA) output was experimentally measured for a 6464 crossbar for VMM and the resulting current from column 1 and column 64 were compared with the expected values using software VMM. Non-idealities in the devices and circuitry result in a drastic difference between the ideal output and the actual output [9] @Copyright 2018 Nature Electronics.

potential in realizing neuromorphic architecture. Various studies have suggested that these emerging devices can allow for the merging of computing and memory, in turn allowing for efficient use of chip area using crossbar arrays. The potential of this type of design is showcased by the dot product engine (DPE) by Hu et al. [12] at Hewlett Packard Labs in 2016. Many neural networks require the use of vector matrix multiplication (VMM), which needs numerous cycles to be computed by traditional CMOS based digital systems. The DPE, however, provides a high-density and energy efficient alternative. Essentially, by using the memristor crossbar array, the dot product can be calculated within one cycle, which makes it a very powerful tool that many neural networks can take advantage of. The so-called DPE is analog by nature, and in an ideal scenario, where the memristive devices used are linear and there are no circuit parasitics, the input voltage signal (V_{in}) would be applied to the rows of the crossbar and would be multiplied by the respective conductance of each memristor following Ohm's law. The resulting currents are then summed across each column following Kirchoff's current law (KCL) which are sensed at all the columns. The entire process occurs within a single time step and the proposed schematic is visualized in Figure 2(a).

While these studies [12,26,51] very well in an ideal situation, in the real world, many non-idealities come into play. Due to the DPE being sensitive to the conductance values of the individual memristive elements, one must account for wire resistance, resistance from input and output stages, as well as the nonlinear current-voltage (I-V) dynamics of contemporary memristive devices and the drift of conductance state that can cause deviations from expected performance, shown in Figure 2(b). The Hewlett Packard group noted that without accounting for all of these variables, the resulting output of the DPE yielded poor accuracy. To combat this, the group invented a conversion algorithm that accounted for device physics and the circuit issues that arise based on device models that were calibrated from real-world, fabricated devices. The algorithm consisted of linearly mapping the desired matrix to an ideal memristive crossbar array to set a benchmark for ideal crossbar behavior. This was followed by simulating the non-ideal crossbar array and tuning the memristor conductions to match the current that passed through each device in the ideal scenario by using a closed-loop tuning scheme to program the non-ideal memristors to the target conductance values set by the ideal crossbar array with a 1% error tolerance. This drastically improved the DPEs performance in accuracy and after it was simulated and implemented with an auto-encoder classifier neural network for pattern recognition tasks using the proposed conversion algorithm, the DPE based NN could perform with 99% accuracy for the MNIST data set with only a 4-bit digital-to-analog converter/analog-to-digital converter (DAC/ADC) requirement [12].

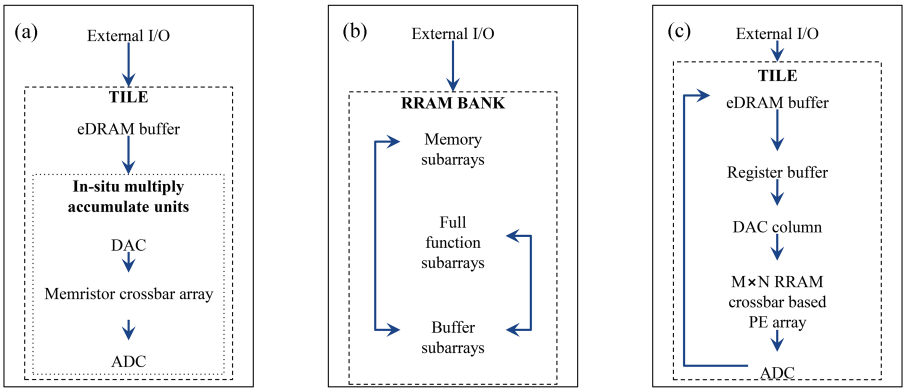

Figure 3 (a) ISAAC architecture using tiles [13]. (b) Simplified design of PRIME [14]. External I/O data is fed into RRAM crossbar based banks. Memory subarrays only store data. Full function subarrays can store data or perform computation, using a CMOS controller to enable configuration. The buffer subarrays serve as data buffers for the full function subarrays. (c) AEPE Implementation with efficient tile architecture [15]. The register buffer feeds into M rows of crossbar based PE arrays through an assigned DAC for each row.

4.1 ISAAC

Currently attempting to integrate emerging memristive devices to help improve the performance of neural networks is a field of intense study. Shafiee et al. [13] developed in-situ analog arithmetic in crossbars (ISAAC), a CNN accelerator architecture capable of holding synaptic weights and performing dot-product operations in the same memristor crossbar. The system, shown in Figure 3(a), is organized into multiple tiles, each containing its own crossbar array. Each crossbar array is dedicated to process a set of neurons that compose a given CNN layer, the outputs of which feed to other crossbars that feed the next layer.

The group proposes using a memristor crossbar array to store input weights, as well as perform dot product operations in an analog manner. Using the crossbar is conducive to the use of synaptic weight matrices that define each CNN layer, which is the dominant data structure in CNNs. Due to the nature of CNNs, relevant algorithms tend to involve large numbers of dot product operations, thus using the memristor crossbar would help reduce the need for certain CMOS based devices, reducing power consumed and the amount of on-chip area required. Along with that, by being intrinsically analog, the memristor crossbar is amenable to analog computations. With that said, however, the implementation proposed here does not allow for crossbars be trained efficiently on the fly, therefore, specific crossbars need to be dedicated to a set of neurons in a given layer, the output of which is fed to other crossbars that process the next layer and so on. This structure is conducive to a pipelined architecture and most of the chip area is dedicated to the dot product engines. On a variety of CNN and deep neural network (DNN) workloads, ISAAC provides 14.8× throughput, 5.5× less energy consumed, and 7.5× in computational density when compared to the DaDianNao architecture.

4.2 PRIME

Chi et al. [14] devised PRIME, a novel Processing-In-Memory solution with architecture that uses resistive random access memory (ReRAM) crossbar arrays to accelerate neural network applications. The crossbars allow for efficient matrix-vector multiplication and allow the merging of computation and memory functions, along with the fact that ReRAM is a good candidate for processing-in-memory thanks to its large capacity, fast read speed, and computation capability among other things. Using a software/hardware interface, PRIME is capable of implementing various neural networks and the architecture allows for dynamic reconfiguration, allowing it to behave as NN accelerators or as normal memory for increased memory space. This morphable functionality had insignificant area overhead. To achieve this, the researchers designed the architecture to enable computation in memory while reusing peripheral circuitry for both memory and computation functions. The architecture was designed to leverage processing in the ReRAM cells, which is achieved by partitioning a ReRAM bank into memory subarrays, full function

subarrays, and buffer subarrays, as shown in Figure 3(b). Memory subarrays are designed to only store data. The full function subarrays can serve as additional memory, or execute NN computation, relying on a CMOS based controller to control the configuration. The buffer subarrays, which are the memory subarrays closest to the full function subarrays, serve as data buffers for the full function subarrays. The study claims that with this implementation, it is able to achieve roughly 2360× the performance and 895× reduced power consumption over leading state-of-the-art CMOS designs, showing the potential of using these emerging devices over pure CMOS implementations. Performance was measured using MiBench, which uses large NNs and requires high memory bandwidth. Compared to a CMOS solution, PRIME showed significantly less execution time to complete the task.

4.3 AEPE

An alternative design that further improved on previous architectures was proposed by Tang et al. [15]. Area efficient and power efficient (AEPE) is a deep CNN accelerator that improves upon ISAAC and PRIME by further minimizing area on chip as well as reducing power consumption, balancing a tradeoff with accuracy. To address the area and power inefficiency in pipelined architecture such as ISAAC, AEPE reduces the required number of DACs with analog data reuse, where only one column of the processing element arrays is assigned a DAC to convert signals. The architecture uses tiles to connect RRAM crossbars with a local connection network on chip (NOC), allowing for more area efficiency than using a shared data bus. As shown in Figure 3(c), each tile is composed of a crossbar processing element array, an embedded dynamic random access memory (eDRAM) buffer, a ping-pong buffer, and a group of DACs. The eDRAM stores the output of the processing element (PE) array and the partial sums of the outputs, which are then sent to the vertical PEs, converted by ADCs to a digital signal in the process. The ping-pong buffer provides input data to the PE array, connecting to the DACs which are used to convert the digital signals to analog inputs, which are then transmitted across horizontal PEs. Each PE is composed of an ADC, an accumulating unit, a shift unit, and two sample-and-hold modules. The design also exploits data reuse in signals, which reduces the total number of DACs, as well as the port width of the buffers used for time multiplexing. The processing units are designed to strike a balancing trade-off by reducing the resolution of the ADCs used and the overall accuracy of the algorithm. This is due to the fact that as the resolution of the ADCs increases, the power consumption increases in an exponential manner. Along with trying to improve power efficiency, they also looked into increasing area efficiency by using local communications among the processing elements to reduce the width of the port in the eDRAM buffer and avoid having to use a high-cost shared bus. In the closing statements, it is mentioned that the AEPE implementation has 2.71× more power efficiency, 2.41× more area efficiency, and less than a 0.5% loss of accuracy when compared to other state-of-the-art RRAM crossbar-based accelerators.

5 Implementations of memristor neural networks

Although memristive devices are still in an incipient stage of development, researchers have already begun to move from the planning and simulation phase to actually fabricating and implementing these emerging devices for various real-world neural network applications. In spite of the fact that these networks are initial experimental demonstrations, they still testify to the potential of these devices as future computing solutions.

5.1 1T1R level-based neural networks

Generally speaking, brain inspired technologies use analog weights to complete tasks efficiently. To achieve this, CMOS approaches depend on on-chip weight storage using SRAM or off-chip weight storage using DRAM. Allocating chip space for on-chip memory is area inefficient and, in turn, limits the total amount of memory available for use by the network. On the other hand, using an off-chip weight storage system based on DRAM significantly increases power consumption and increases overall latency as data

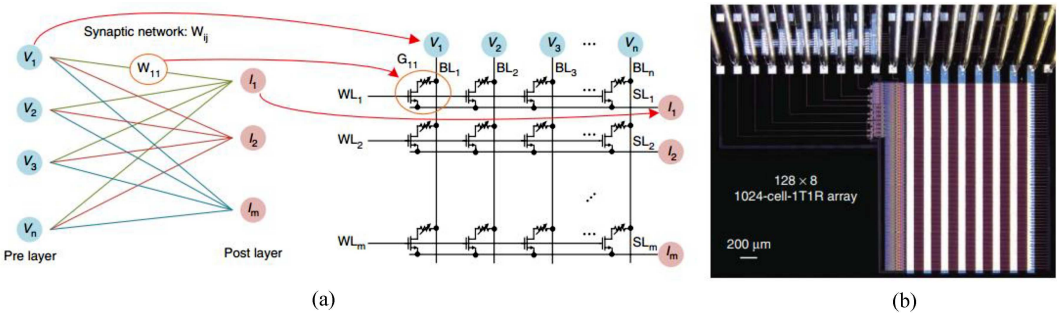

Figure 4 (Color online) (a) A single-layer is mapped onto the one transistor one memristor (1T1R) array; (b) micrograph of the 1024 cell 1T1R array fabricated using CMOS compatible processes [16] @Copyright 2017 Nature Communications.

has to be transferred from the off-chip memory to the processor. An alternative to this is to program the analog weights as specific conductance values in memristive devices. The one transistor one memristor (1T1R) structure allows for precise conductance tuning of each memristor in the crossbar array allowing each individual memristor to be tuned using the transistor's gate linear I-V relation [9]. In addition to that, the structure has the added benefit of mitigating sneak path current in larger crossbars, as well as limiting the compliance current to prevent damage to the memristive devices. Yao et al. [16] proposed and fabricated a neuromorphic network composed of a 1024-cell (128 × 8 array) for grey-scale face classification from the Yale Face Database using analog, nonvolatile memristive devices as electronic synapses. They were able to create a network that could be trained online and used less than 20 times the energy required for an Intel Xeon Phi processor utilizing hypothetical on-chip digital resistive RAM. Their optimized memory cell structure is compatible with the traditional CMOS process and allows for bidirectional analog behavior and is claimed to be capable of running more complex, deep neural networks than what was demonstrated at the time of the study.

The network, visualized in Figure 4(a), was a single layer perceptron neural network using 1T1R as the synapse with two programming schemes: One using a write-verify method for classification performance and one without the write-verify method to simplify the required control circuitry. During the write-verify programming scheme, identical voltage pulses were applied to a cell to modulate its conductance. Without the write-verify programming scheme, only one SET or RESET pulse was applied without checking if the target conductance was reached. The authors noted that while this does help to simplify the control system, it slows down convergence. The 1T1R synapse design allows for bidirectional switching performance and uniformity and also helps eliminate sneak path currents.

To train the network, three images of three different people were presented at the input of the network. By measuring the resulting total current, a weighted sum can be acquired, and by applying the sum to the tanh function, 3 output values were obtained. The neuron with the largest output value was what was used to classify the images. The network was tested at varying initial conductance distribution states with unseen face images, some with up to 31.25% noise and the researchers were able to conclude that their network, in regards to recognition rate, performed similarly to standard computing systems, which have a rate of ∼91.48%. The write-verify programming scheme allowed for an accuracy of 88.08% within 10 iterations, and even without the write-verify programming scheme, a recognition rate of 85.04% was achieved, granted more time was required for convergence. They also stated that the initial conductance distribution did not have much of an effect on the network overall regardless of programming scheme. Furthermore, while the write-verify scheme did require more programming pulses and had a longer single iteration time than without the write-verify scheme, it showed to have better performance in energy consumption, time to convergence, and recognition accuracy than without it.

Li et al. [9] also utilized the 1T1R structure in allow for analog signal and image processing using a 128 × 64 crossbar array. By using a high yield (99.8% device yield), reconfigurable Ta/HfO2 memristor crossbar on top MOS transistors (Figure 5), VMM was computed, allowing for efficient performance to filter, analyze, compress, and encode data. The setup was used to demonstrate 64 levels of conductance

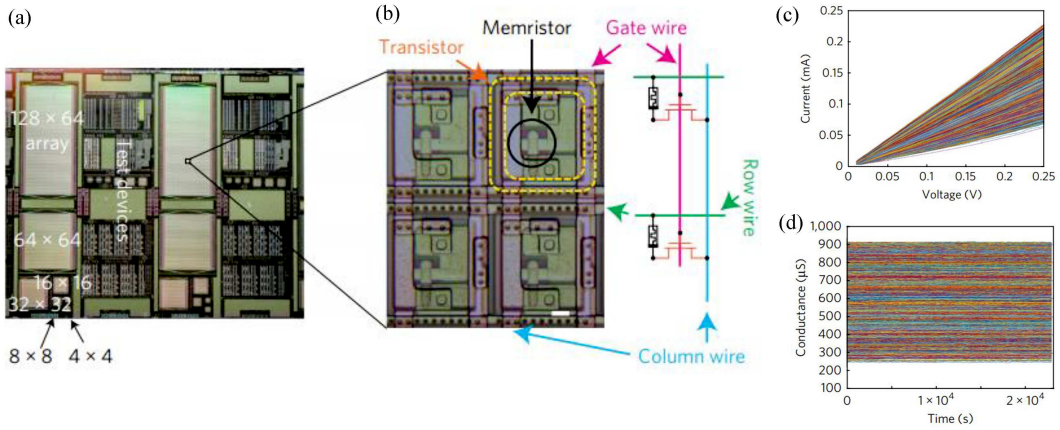

Figure 5 (Color online) (a) Photograph of fabricated 1T1R crossbars. Two dies are shown, each one containing various array sizes from 4×4 to 128×64 cells. (b) Micrograph of four cells in a 1T1R array (scale bar, 10 mm). (c) Relatively linear I-V curves for all the devices over the chosen conductance range. (d) Device state retention and read disturbance (1000 cycles of 0.2 V read pulses) at room temperature show no discernible drift [9] @Copyright 2017 Nature Electronics.

(6 bits of digital precision), which is suitable for many tasks involving machine learning algorithms.

For analog signal processing and image compression, the array was configured to implement a discrete cosine transformation (DCT). The values of the DCT transform were mapped as the conductances of the memristors using a linear transform accounting for negative values. A 64 × 64 matrix of synapses was mapped with the DCT matrix values and the output accuracy analyzed by measuring the experimentally measured currents which were compared to the expected currents for each column given a range of inputs and high agreement between the two was observed. The setup was then configured with a 1D DCT to be used as a spectrum analyzer, wherein sine waves with different frequencies were inputted and the spectrum output displayed the frequency spectrum, showing good agreement with a software DCT. Image compression was implemented using a 2D DCT, where the input pixel intensities were converted to voltages and applied to the crossbar array row by row, then column by column. The output was the cosine transforms of the input images, which were then reconstructed using the inverse DCT. With only 1/20th of the original image, it was still possible to reconstruct a reasonable image. Next, a 2D convolutional image filter was demonstrated to reduce noise and locate edges of a given image, a frequent step of the first layer of many CNNs. The image intensities were converted to voltages that were applied to the rows of the crossbar and each pixel of the filtered image was generated by the dot product of the 25-dimensional voltage vector mapped from a 5×5 input sub-image and a 25-dimensional conductance vector mapped from a 5×5 convolution matrix.

This implementation has the added benefit of being able to process analog signals directly from the sensors without any need for ADCs, further decreasing power consumption. The crossbar was composed of over 8000 memristive devices and though most of them functioned as expected, it was noted that unresponsive devices, especially devices stuck in the high conductance state had an adverse effect on the overall accuracy of the output as well as a power consumption. The concluding remarks stated that this implementation allows for roughly 119.7 trillion operations per second per watt.

5.2 Biologically plausible networks

To go along with the implementations, various biologically plausible algorithms have also been proposed. A more biologically realistic network was created in a recent study by Pedretti et al. [6], who demonstrated an SNN with memristive synapses whose weights were updated via STDP during unsupervised learning. The study used a 1T1R configuration for a synapse, relying on two states (high resistance state and low resistance state) to learn and recognize patterns. The group demonstrated unsupervised learning of static patterns and adaptation to a dynamic pattern using a perceptron-like network of memristive synapses where weights updated by local STDP via real time feedback using the schematic shown in Figure 6(b).

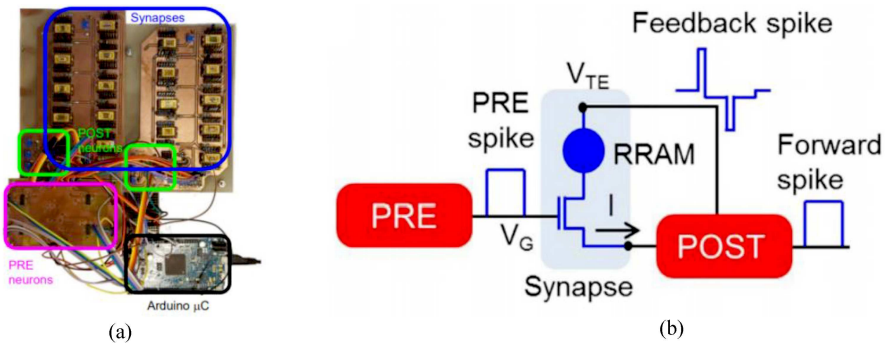

Figure 6 (Color online) (a) Synapse array with peripheral circuitry; (b) a pre-synaptic device controls the FET gate of the 1T1R structure, while the post-synaptic device receives the input current and controls the synapse top electrode to induce synaptic current and stimulate synaptic potentiation or depression during the fire [6] @Copyright 2017 Scientific Reports.

By applying a pre-synaptic spike to the gate terminal of the transistor in the 1T1R synapse, the resulting output current from the synapse can be integrated via an integration stage of a leaky-integrate and fire (LIF) post-synaptic neuron as a corresponding internal voltage. Once the internal voltage accumulated to be larger than a predefined threshold voltage, a forward spike would be generated by the post-synaptic neuron and a feedback spike of positive and negative pulses would backpropagate to the synapse for STDP, following the rule that a post spike following a pre-spike allows for synaptic potentiation by enabling the transistor in a given synapse with the pre-spike during a positive spike from the feedback of the post-synaptic neuron. Synaptic depression was achieved if the post-synaptic neuron fired a spike before the pre-synaptic spike was received, which resulted in enabling the transistor in a given synapse during a negative spike from the feedback of the post-synaptic neuron.

Using this SNN, the group created a two layer perceptron network that allowed for pattern recognition. The network was composed of sixteen synapses connecting a single post neuron which was responsible for recognition and classification of inputted patterns. To train the network, visual patterns were randomly submitted to synapses to induce STDP. Training was considered successful if the post neuron only fired when presented with the same pattern used during training and no other patterns incited a response. The group was also able to enable multiple pattern learning with a second post neuron which allowed for lateral inhibition among the post neurons. By having the firing post neuron send an inhibiting spike that reduced the internal voltage of the other post neuron, the active post neuron could effectively inhibit the activation of the other, preventing scenarios in which both fire at the same time, which is undesirable if each neuron is used to recognize a specific pattern.

To further emulate neuronal behavior at a more fundamental level, Wang et al. [17] fabricated and experimentally demonstrated a fully integrated memristive neural network, using the schematic in Figure 7. The network featured an 8×8 crossbar of 1T1R devices based on drift memristors that served as the weighted synapses that fed into eight diffusive memristor [11] neurons, shown in Figure 8. The diffusive memristors depend on a conductive Ag nanoparticle bridge that forms between the top and bottom electrode under voltage bias to achieve a low resistance state. Without the voltage bias, however, the conductive bridge naturally dissolves with time, allowing the device to return to a high resistance resting state. The volatile Ag based diffusive dynamics of the diffusive memristors allowed them to implement a leaky integrate and fire model since the diffusive memristor neuron would return to a resting state after it fired or if the stimulation interval was exceeded without reaching the neuron's threshold. Each artificial neuron was designed to be simplistic and allow for further scalability all while maintaining biological plausibility: The threshold behavior of the diffusive memristor is comparable to the biological ion channels embedded within the membrane of neurons and the integrate and fire behavior of the artificial neurons was controlled using a resistor-capacitor (RC) circuit that mimicked the biological axial resistance and the membrane capacitance. It was also noted that the threshold has an associated probability distribution function that resembles the stochastic behavior commonly observed in biological neurons.

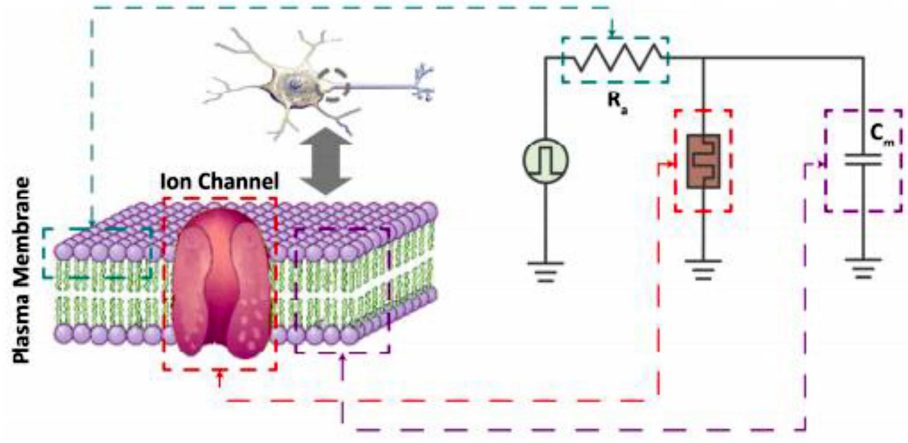

Figure 7 (Color online) Diffusive memristor neuron schematic. The diffusive memristor behaves in a fundamentally similar manner as a biological ion channel on the soma membrane of a neuron [17] @Copyright 2018 Nature Electronics.

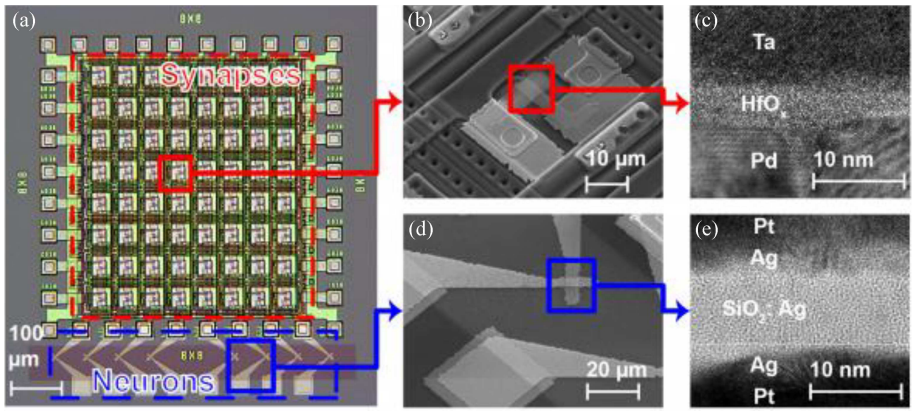

Figure 8 (Color online) (a) All memristor network fabricated using (b) and (c) drift memristors for synapses and (d) and (e) diffusive memristors for neurons [17] @Copyright 2018 Nature Electronics.

The resulting fully connected network was demonstrated to be useful as a convolutional layer for pattern classification and, with some software pooling and signal conversion, a rectified linear unit (ReLU). The synapses were trained using unsupervised learning which depended on a simple STDP training scheme that relied on the firing time and pulse width of the output of the eight diffusive memristor neurons. Furthermore, lateral inhibition was utilized to improve the discrimination of input patterns and increase overall network energy efficiency and was achieved by increasing the voltage of the input pattern with a slow ramping rate until a neuron fired to limit the number of simultaneously firing neurons. The study concluded with the claim that the presented network is one the simplest, yet biologically faithful demonstrations to date.

Another example is sparse coding, wherein the algorithm attempts to the find optimal signal representation by reducing the complexity of incoming signals. This allows for a more efficient representation of information, allowing for improved data analysis due to the network being able to identify otherwise hidden features of the incoming stimulus, which is useful in pattern recognition and feature extraction tasks. Sparse coding with a memristor network was demonstrated by Sheridan et al. [18], which utilized a 32×32 memristor crossbar array to implement the algorithm and perform image analysis using learned dictionaries. The prototype worked in two phases: The first phase was to code learned dictionaries, where the dictionary was directly mapped element-wise onto the memristive crossbar as the conductance values. To implement an input vector (such as pixel intensities of an image), pulses of a fixed amplitude and varying widths were used corresponding to the input data value. The resulting charge passed by a memristor is the product of the input data and the dictionary weight of that element. In the same cycle,

Table 1 Summary of experimental memristor based network implementations presented in literature[a)]

Implementation	Computing efficiency	Accuracy	Power consumption	Memristor used/Dielectric thickness	Crossbar cell count	Reference
1T1R face classifying perceptron network	Converges within 10 iterations	88% recognition accuracy within 10 iterations	∼30 nJ per epoch for classification task	Memristor-8 nm $HfAl_yO_x$	128×8 cells	[16]
1T1R Ta/HfO_2 memristor network	119.7 trillion operations per second per watt	91.71% recognition accuracy on MNIST data set	13.7 mW for image compression	Memristor-5 nm HfO_2	128×64 cells	[9, 19, 20]
1T1R STDP network	–	–	–	Memristor-10 nm HfO_2	4×4 pre-cells and 1 post-cell	[6]
Diffusive memristor artificial neuron network	–	–	–	Diffusive memristor-10 nm Ag/SiO_2 Drift Memristor-5 nm HfO_2	8×8 synaptic crossbar with 8 neurons	[17]

a) While the 1T1R networks did utilize CMOS processes, specific relevant details were not always provided. For the 1T1R STDP network, the recognition threshold was set so that no false-positives were reported.

all elements sharing a column have their currents summed following KCL. The value of the preceding single-step dot product is then added to the membrane potential of the neuron and if the potential is greater than a preset threshold, the neuron is set as active for the next phase. In the second phase, the input image is reconstructed according to the currently active neurons and compared with the original input image.

Table 1 summarizes and compares the various aforementioned implementations on performance and crossbar size for their respective tasks. While all of these networks are prototypes, they show promising results to reduce power consumption, while simultaneously allowing for deeper scalability and increased computing efficiency.

6 Conclusion and outlook

Neural networks derived from the biological brain have a lot of potential as powerful and energy-efficient computing tools in today's era of Big Data and the IoT. The aforementioned implementations with emerging devices, although still in experimental phases, are able to handle their assigned tasks in a much more efficient manner than their state-of-the-art CMOS counterparts, both in terms of execution time and energy, clearly showing the advantages these novel networks based on emerging devices have over traditional CMOS based systems. Currently, however, many of these implementations still depend on additional circuitry, such as control logic and ADCs/DACs to be able to complete a given task, which in turn increases power consumption and adds to the circuit level complexity of these networks. These studies just scratch the surface of what is possible as emerging devices continue to be researched and improved upon in the coming decades and substantial improvements in network design and algorithms will be required to get the most out of these upcoming devices.

While the ultimate goal is to use emerging devices to implement advanced NNs, pure CMOS based implementations can be used currently to demonstrate proof of concept and the benefits of using and optimizing NNs while emerging devices will be able to improve upon that even further. Critical device properties for use in memristor based neural networks include predictable I-V relationships, symmetric

responses to applied pulses with low cycle to cycle variation, and minimal device to device variation. Although a number of these properties currently present issues to be worked out with modern memristors, while these devices are still in development it is useful to use CMOS based NN implementations to understand and further improve the architecture and algorithms for NNs. These improvements can later be translated to networks based on emerging devices, which will in turn allow for improved power and computation efficiency. The final goal is to construct an implementation that not only has low energy requirements, but also allows for high speed computation to be carried out in an area efficient manner.

Though our knowledge of neuroscience is ever expanding, what we have been able to learn from research conducted in that domain has been useful in providing a basis for studies and experiments done using emerging electronic devices to allow for constant improvements and revisions of current neuromorphic circuits and networks. Further understanding of the brain and the units it is comprised of, including the neuron and synapse, will be key to ushering in a new age of brain-inspired computing that will provide useful solutions for many of the problems with today's computing, including the von Neumann bottleneck and device scaling. As of now, many NNs aim to perform a specific function, unlike the brain which handles numerous processes simultaneously. Research needs to continue at the device and network level to allow for us to get closer to achieving brain-like computation. While we might not be able to mimic biology perfectly with current emerging devices, memristors in particular do show a lot of promise in the fact that they are able to mimic certain vital aspects of the nervous system, including basic neuron and synapse functionality, bringing us closer to achieving a true "thinking machine".

References

1 Mack C A. Fifty years of Moore's law. IEEE Trans Semicond Manufact, 2011, 24: 202–207
2 Schulz M. The end of the road for silicon? Nature, 1999, 399: 729–730
3 von Neumann J. First draft of a report on the EDVAC. IEEE Ann Hist Comput, 1993, 15: 11–21
4 Anderson H C. Neural network machines. IEEE Potentials, 1989, 8: 13–16
5 Squire L R, Berg D, Bloom F, et al. Fundamental neuroscience. Curr Opin Neurobiol, 2008, 10: 649–654
6 Pedretti G, Milo V, Ambrogio S, et al. Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity. Sci Rep, 2017, 7: 5288
7 Zamarreño-Ramos C, Camuñas-Mesa L A, Pérez-Carrasco J A, et al. On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. Front Neurosci-Switz, 2011, 5: 1–22
8 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444
9 Li C, Hu M, Li Y, et al. Analogue signal and image processing with large memristor crossbars. Nat Electron, 2018, 1: 52–59
10 Hebb D O. The first stage of perception: growth of the assembly BT — the organization of behavior. Organ Behav, 1949, 4: 60–78
11 Wang Z, Joshi S, Savel'ev S E, et al. Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. Nat Mater, 2017, 16: 101–108
12 Hu M, Strachan J P, Li Z, et al. Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication. In: Proceedings of the 53rd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, 2016. 1–6
13 Shafiee A, Nag A, Muralimanohar N, et al. ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In: Proceedings of ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), Seoul, 2016. 14–26
14 Chi P, Li S, Xu C, et al. PRIME: a novel processing-in-memory architecture for neural network computation in reRAM-based main memory. In: Proceedings of ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), Seoul, 2016. 27–39
15 Tang S, Yin S, Zheng S, et al. AEPE: an area and power efficient RRAM crossbar-based accelerator for deep CNNs. In: Proceedings of IEEE 6th Non-Volatile Memory Systems and Applications Symposium (NVMSA), Hsinchu, 2017. 1–6
16 Yao P, Wu H, Gao B, et al. Face classification using electronic synapses. Nat Commun, 2017, 8: 15199
17 Wang Z, Joshi S, Savel'ev S, et al. Fully memristive neural networks for pattern classification with unsupervised learning. Nat Electron, 2018, 1: 137–145
18 Sheridan P M, Cai F, Du C, et al. Sparse coding with memristor networks. Nat Nanotech, 2017, 12: 784–789
19 Hu M, Graves C E, Li C, et al. Memristor-based analog computation and neural network classification with a dot product engine. Adv Mater, 2018, 30: 1705914
20 Li C, Belkin D, Li Y, et al. Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. Nat Commun, in press, 2018
21 Zarudnyi K, Mehonic A, Montesi L, et al. Spike-timing dependent plasticity in unipolar silicon oxide RRAM devices. Front Neurosci, 2018, 12: 57

22 Yu S M, Li Z W, Chen P-Y, et al. Binary neural network with 16 Mb RRAM macro chip for classification and online training. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), San Francisco, 2016. 1–4
23 Prezioso M, Merrikh-Bayat F, Hoskins B D, et al. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. Nature, 2015, 521: 61–64
24 Yu S. Neuro-inspired computing with emerging nonvolatile memorys. Proc IEEE, 2018, 106: 260–285
25 Alibart F, Zamanidoost E, Strukov D B, et al. Pattern classification by memristive crossbar circuits using ex situ and in situ training. Nat Commun, 2013, 2013: 2072
26 Merrikh B F, Prezioso M, Chakrabarti B, et al. Advancing memristive analog neuromorphic networks: increasing complexity, and coping with imperfect hardware components. ArXiv: 1611.04465
27 Du C, Cai F, Zidan M A, et al. Reservoir computing using dynamic memristors for temporal information processing. Nat Commun, 2017, 8: 2204
28 Mehonic A, Kenyon A J. Emulating the electrical activity of the neuron using a silicon oxide RRAM cell. Front Neurosci-Switz, 2016, 10: 57
29 Chen Y J, Luo T, Liu S L, et al. DaDianNao: a machine-learning supercomputer. In: Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture, Cambridge, 2014. 609–622
30 Merolla P A, Arthur J V, Alvarez-Icaza R, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science, 2014, 345: 668–673
31 Jouppi N P, Young C, Patil N, et al. In-Datacenter performance analysis of a tensor processing unit. In: Proceedings of the 44th Annual International Symposium on Computer Architecture, Toronto, 2017. 1–12
32 Gawande N A, Landwehr J B, Daily J A, et al. Scaling deep learning workloads: NVIDIA DGX-1/Pascal and intel knights landing. In: Proceedings of 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Lake Buena Vista, 2017. 399–408
33 Chua L O. Memristor — the missing circuit element. IEEE Trans Circuits Syst, 1971, 18: 507–519
34 Strukov D B, Snider G S, Stewart D R, et al. The missing memristor found. Nature, 2008, 453: 80–83
35 Yang J J, Pickett M D, Li X, et al. Memristive switching mechanism for metal/oxide/metal nanodevices. Nat Nanotech, 2008, 3: 429–433
36 Yang J J, Miao F, Pickett M D, et al. The mechanism of electroforming of metal oxide memristive switches. Nanotechnology, 2009, 20: 215201
37 Chua L. Resistance switching memories are memristors. Appl Phys A, 2011, 102: 765–783
38 Yang J J, Strukov D B, Stewart D R. Memristive devices for computing. Nat Nanotech, 2013, 8: 13–24
39 Yang J J, Williams R S. Memristive devices in computing system: promises and challenges. ACM J Emerg Tech Com, 2013, 9: 1–20
40 Pickett M D, Strukov D B, Borghetti J L, et al. Switching dynamics in titanium dioxide memristive devices. J Appl Phys, 2009, 106: 074508
41 Alibart F, Gao L, Hoskins B D, et al. High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. Nanotechnology, 2012, 23: 075201
42 Choi S, Sheridan P, Lu W D. Data clustering using memristor networks. Sci Rep, 2015, 5: 10492
43 Yang J J, Zhang M X, Strachan J P, et al. High switching endurance in TaO_x memristive devices. Appl Phys Lett, 2010, 97: 232102
44 Choi B J, Torrezan A C, Strachan J P, et al. High-speed and low-energy nitride memristors. Adv Funct Mater, 2016, 26: 5290–5296
45 Yoon J H, Zhang J, Ren X, et al. Truly electroforming-free and low-energy memristors with preconditioned conductive tunneling paths. Adv Funct Mater, 2017, 27: 1702010
46 Li C, Han L, Jiang H, et al. Three-dimensional crossbar arrays of self-rectifying $Si/SiO_2/Si$ memristors. Nat Commun, 2017, 8: 15666
47 Shulaker M M, Hills G, Park R S, et al. Three-dimensional integration of nanotechnologies for computing and data storage on a single chip. Nature, 2017, 547: 74–78
48 Yu S, Wu Y, Jeyasingh R, et al. An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. IEEE Trans Electron Devices, 2011, 58: 2729–2737
49 Chang T, Yang Y, Lu W. Building neuromorphic circuits with memristive devices. IEEE Circuits Syst Mag, 2013, 13: 56–73
50 Chang T, Jo S H, Kim K H, et al. Synaptic behaviors and modeling of a metal oxide memristive device. Appl Phys A, 2011, 102: 857–863
51 Gaba S, Sheridan P, Zhou J, et al. Stochastic memristive devices for computing and neuromorphic applications. Nanoscale, 2013, 5: 5872–5878