PROCEEDINGS OF THE IEEE

Neuro-Inspired Computing with Emerging Non-Volatile Memory

Journal:	*Proceedings of the IEEE*
Manuscript ID	0277-REG-2017-PIEEE
Manuscript Categories:	Regular Paper
Date Submitted by the Author:	02-Nov-2017
Complete List of Authors:	Yu, Shimeng; Arizona State University,
Keyword:	

Neuro-Inspired Computing with Emerging Non-Volatile Memory

Shimeng Yu

School of Electrical, Computer and Energy Engineering, Arizona State University

Email: shimengy@asu.edu

Abstract:

This comprehensive review summarizes state-of-the-art, challenges and prospects of the neuro-inspired computing with emerging non-volatile memory devices. First, we discuss the demand for developing neuro-inspired architecture beyond today's von-Neumann architecture. Second, we summarize the various approaches to designing the neuromorphic hardware (digital vs. analog, spiking vs. non-spiking, online training vs. offline training) and discuss why emerging non-volatile memory is attractive for implementing the synapses in the neural network. Then, we discuss the desired device characteristics of the synaptic devices (e.g. multilevel states, weight update nonlinearity/asymmetry, variation/noise), and surveyed a few representative material systems and device prototypes reported in the literature that show the analog conductance tuning. These candidates include phase change memory, resistive memory and ferroelectric memory and floating-gate transistors, etc. Next, we introduce the crossbar array architecture to accelerate the weighted sum and weight update operations that are commonly used in the neuro-inspired learning algorithms, and review the recent progresses of array-level experimental demonstrations for pattern recognition tasks. In addition, we discuss the peripheral neuron circuit design issues and present a device-circuit-algorithm co-design methodology to evaluate the impact of non-ideal device effects on the system-level performance (e.g. learning accuracy). Finally, we give an outlook on the customization of the learning algorithms for efficient hardware implementation.

Keywords:

Neuromorphic computing, neural network, machine learning, hardware accelerator, non-volatile memory, resistive memory, synaptic device

1. Introduction

Artificial intelligence (AI) that allows machines to think and act like human beings is reviving, which is a hot topic today not only in academia but also has made remarkable social impact (e.g. Google's AlphaGo [1]). In recent years, artificial neural networks (i.e. machine/deep learning) has shown significantly improved accuracy in large-scale visual/auditory recognition and classification tasks, some even surpassing human-level accuracy [2]. In particular, convolutional neural network (CNN) [3] and recurrent neural network (RNN) [4] algorithms and their variants have proved their efficacy in a wide range of image, video, speech, and biomedical applications. To achieve incremental accuracy improvement, state-of-the-art deep learning algorithms tend to aggressively increases the depth and size of the neural network. For example, Microsoft's Residual-Net (which won the ImageNet 2015 image classification competition [5]) has more than one hundred of layers [6]. This poses significant challenges for hardware implementations in terms of computation, memory, and communication resources. For example, Google's stacked autoencoder algorithm was able to successfully identify faces of cats from 10 million random images taken from YouTube videos [7]. Yet this task was accomplished on a cluster of 16,000 processor cores consuming ~100 kW power and used three days to train the network.

Today's deep learning is typically trained with graphic processing unit (GPU) accelerators on the data center or cloud side. Specific designed accelerators such as Manchester's SpiNNaker [8], Heidelberger's BrainScaleS [9], and Google's tensor processing unit (TPU) [10] have been developed to run large-scale neuromorphic and/or deep learning algorithms. On the embedded system or Internet of Tings (IoT) edge computing side, such as autonomous driving, smart sensors and wearable devices, severe constraints exist in performance, power and area. Several application-specific integrated circuits (ASIC) on-chip solutions in silicon complementary-metal-oxide-semiconductor (CMOS) technology such as IBM's TrueNorth [11], MIT's Eyeriss [12] and a series of CNN accelerators [13, 14, 15] have been developed. However, limitations still exist on on-chip memory capacity, off-chip memory access, and online learning capability. In particular, the CMOS ASIC designs show that on-chip memory is the biggest bottleneck for energy-efficient real-time computing, which means storing millions of parameters and loading/communicating them to the place where computing actually occurs. Today's neuromorphic chips or ASIC accelerators typically utilize static random access memory (SRAM) as the synaptic memory on-chip. Although SRAM technology has been following the CMOS scaling trend well, the SRAM density (100-200 F^2 per bit cell, F is the technology node) and on-chip SRAM capacity (typically a few MB) are insufficient for storing the extremely large number of parameters in deep learning algorithms (typically hundreds of MB). Leakage current is undesirable, and parallelism is limited due to the row-by-row operation in the digital SRAM array.

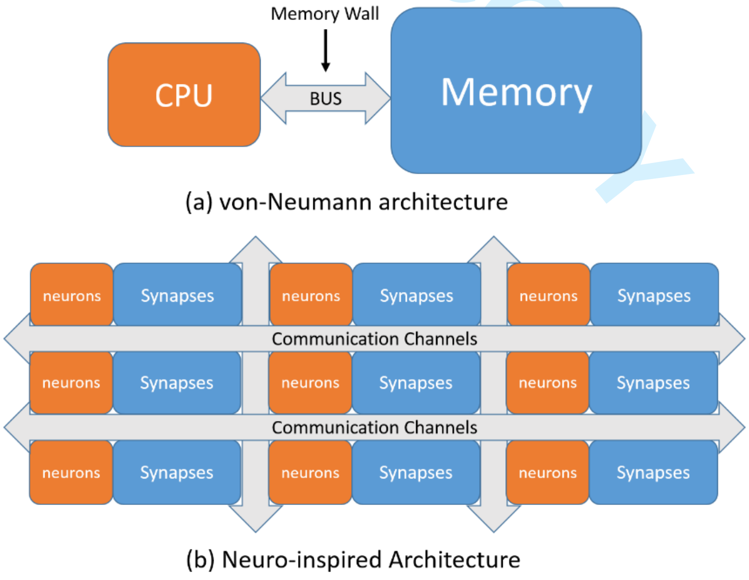
As an alternative hardware platform, emerging non-volatile memory (eNVM) devices have been proposed for on-chip weight storage with higher density (typically 4-12 F^2 per bit cell) and fast parallel analog computing with low leakage power consumption [16]. A special subset of eNVM devices that show multilevel resistance/conductance states could naturally emulate synaptic device in the neural network, namely *resistive synaptic devices* [17]. Examples of resistive synaptic devices include the two-terminal eNVMs such as phase change memory (PCM), resistive random access memory (RRAM) and the three-terminal ferroelectric transistor and floating-gate memory (with analog threshold voltages). The parallelism of resistive crossbar array for matrix-vector multiplication (or dot product) further enables significant acceleration of core neural computations (i.e. weighted sum). A recent analysis by IBM showed that fully-connected multi-layer perceptron (MLP) can be potentially trained faster with lower power consumption with PCM based accelerators than with the conventional GPUs [18]. With optimized device specifications, the eNVM based accelerators could potentially outperform the silicon CMOS ASIC based ones with SRAM synaptic arrays [19].

In the past few years, the research on eNVM based synaptic devices and its integration to the array-level has made remarkable milestones in the past few years. At the device-level, many resistive synaptic device candidates that are capable of tens to hundreds levels of conductance states have been demonstrated at single device level. The resistive synaptic devices could emulate the biological synapse in the sense that ions or atomic migration/rearrangement in the solid-state dielectrics (e.g. in oxides/chalcogenides) could modulate the conductance between the two electrodes, as the biological synapse modulate its conductance via the activation of voltage-gated calcium channels. At the array-level, there have been a few experimental demonstrations of simple neural network algorithms on small-scale (e.g. 12×12) to medium-scale (e.g. 256×256) with software and/or off-chip control. These demonstrations show the great promises for future large-scale integration and prototypes with CMOS on-chip control. In addition, the computer aided design (CAD) or electronic design automation (EDA) tool development has facilitated the co-optimization of device properties with circuits/architectures and algorithms, to address the design challenges associated with device yield, device variability, and array parasitics when the array size is scaled up. Pioneering simulation frameworks have been developed to evaluate the impact of device-level non-idealities (limited weight precision, weight update non-linearity/asymmetry, variation/noise, etc.) on the trade-offs between learning accuracy and training speed/energy.

In this context, it is timely to have a holistic review of the recent progresses in the field of neuro-inspired computing with eNVMs. There are several comprehensive reviews on eNVMs for digital memory applications [20, 21, 22, 23, 24] and CMOS based neuromorphic circuits [25, 26, 27]. There are also pioneering reviews on the synaptic devices which mostly focused on the material aspects of synaptic devices [17, 28, 29]. To our best knowledge, there is no dedicated comprehensive review on eNVMs for neuro-inspired computing hierachically from device-level, array-level up to circuit/architecture-level. In the past few years, significant progresses have been made on the array-level demonstration, and CAD/EDA tool development as discussed above, while these new results have not been reviewed before. With these considerations, we aim to have this review paper to survey state-of-the-art synaptic device properties, small-scale to medium-scale array integration, and early exploration of device-circuit-architecture-algorithm co-design, with the hope of inspiring the research community for the future interdisciplinary collaborations on this emerging and exciting research topic. It should be pointed out that this review is oriented towards using eNVM based devices for energy-efficient computing, instead of emulating the biologically realistic behaviors.

2. Overview of Neuromorphic Hardware Design Approaches

In the conventional von-Neumann computer architecture, the well-known “memory wall” problem that the data movement between the microprocessor and off-chip memory/storage has become the bottleneck of the entire system [30]. This problem becomes even more severe when the large amount of data is required for computation in the training and/or testing of the large-scale neural network. As the neuro-inspired learning algorithms extensively involve large-scale matrix operations, computing paradigms that take advantage of the parallelism at finer-grain level directly on-chip are attractive. One promising solution is the neuro-inspired architecture that leverages the distributed computing in the neurons and localized storage in the synapses .The neuro-inspired architecture leverages the distributed computing in the neurons and localized weight storage in the synapses [31]. Figure 1 shows such revolutional shift of the computing paradigm from the computation-centric (von-Neumann architecture) to the data-centric (neuro-inspired architecture). The neurons are simple computing units (for nonlinear activation of thresholding function) and the synapses are local memories that are massively connected via the communication channels. The ultimate goal of the hardware implementation of the neuro-inspired computing is to supplement (but not supplant) today’s von-Neumann architecture for application-specific intelligent tasks such as image/speech recognition, autonomous driving, etc.



(a) von-Neumann architecture

(b) Neuro-inspired Architecture

Figure 1 A revolutional shift of the computing paradigm from the computation-centric (von-Neumann architecture) to the data-centric (neuro-inspired architecture).

Different hardware platforms with partial parallelism have been explored so far for implementing neuro-inspired learning algorithms. Generally, there are two design approaches (or philosophies) for neuromorphic hardware depending on how to encode the information. The first approach stays on the digital (non-spiking) implementation of machine/deep learning or artificial neural network (ANN) while takes the inspirations from the neural system to maximize the parallel or distributed computation. In the digital implementations, the neuron values are encoded by binary bits or number of pulses. As off-the-shelf technologies, GPUs [32] or field-programmable-gate-arrays (FPGAs) [33] have been widely used for hardware acceleration for machine/deep learning. To further improve the energy-efficiency, CMOS based ASIC accelerators [10] [12] [13, 14, 15] have been prototyped. For example, Google used their custom designed TPU platform to accelerate the complex intelligent computation tasks behind AlphaGo [34]. The digital (non-spiking) approach aims to improve the computation efficiency in terms of the performance per second per watt (e.g. in the metric of operations per second per watt). The second approach exploits the spiking behavior of spiking neural network (SNN) which aims to emulate the biologically realistic neural network more closely. In the spiking approach, the neuron values are encoded by the spiking timing (e.g. the interval between spikes) or even the spike's actual waveform shape. Examples include custom designed CMOS based neuromorphic chips (i.e. Heidelberg's BrainScaleS [9], IBM's TrueNorth [11], etc.). The BrainScaleS platform is based on HICANN chip in 180 nm node that use analog neurons similar as the leaky integrate-and-fire model and digital synapses made of 4-bit 6-transistor SRAM cells and 4-bit digital-analog-converter (DAC) to interface with analog neurons [35]. One die consists of 512 neurons and 100 kilo synapses, and one wafer consists of ~200 kilo neurons and ~40 million synapses. BrainScaleS could run 10 000 × faster than the biological real time (~kHz) but consume 500W/wafer. The TrueNorth chip uses digital neurons and digital synapses made of 1-bit transposable 8-transistor SRAM cell. In particular, one TrueNorth chip integrates 4,096 neuro-synaptic cores with 1 million digital neurons and 256 million SRAM synapses that was fabricated in 28 nm node. The TrueNorth chip demonstrated 70 mW power consumption to perform real-time (30 frames per second) object recognition with very low clock frequency (~kHz).

Table 1 summarizes the categories of different design approaches for hardware implementation of neuro-inspired computing. Here the categories are "loosely" classified based on how the information is encoded and the technological choice of the hardware platforms. The neuron could be encoded either by the digital representation using binary bits or number of pulses or by the spike representation, while the synapses can be either binary or multilevel (in an analog fashion).

Depending on how the training of the neural network is completed, there are two ways of training: offline (ex-situ) training and online (in-situ) training. Offline training means that the training is done by software and the trained weights are loaded to the synaptic arrays of the neuromorphic hardware by one-time programming and then only the inference or classification is performed on the hardware. For example, the TrueNorth supports only offline training (the weights need to be pre-trained and loaded to SRAM arrays). Therefore, such inference-only engine could be used for the edge devices where the model is pre-defined by the cloud, but it could not adapt to the constantly changing input data and/or learn new features during the run-time. Online training means the training is done during runtime on the neuromorphic hardware (i.e. weights are trained on-the-fly). To accelerate the training on the neuromorphic hardware is a much more challenging task. The weight updating rule is different in the machine/deep learning and in the spiking neural network. In the machine/deep learning, typically back-propagation (i.e. by stochastic gradient descent method) layer by layer is used to optimize the objective cost function by comparing error between the prediction and the true label, thus it is a supervised and global training method. By contrast, in the spiking neural network, the local synaptic plasticity (i.e. between neighboring neurons) is often used in an unsupervised fashion. One important biologically-plausible learning rule is the spike-timing-dependent plasticity (STDP) [36]. The STDP learning rule states that if the pre-synaptic neuron fires earlier than the post-synaptic neuron, the conductance of the synapse (weight) will decrease, and vice versa. The change of the weight is larger when the timing between the two neurons firing is closer. However, how to exploit such STDP learning rule (unsupervised and local to two adjacent neurons) to efficiently update the entire neural network remains to be explored. So far, the learning accuracy of machine/deep learning with backpropagation for solving today’s practical classification problems (e.g. image/speech recognition) is significantly better than that of spiking neural network with STDP learning. Therefore, in this review, we focus more on the design perspectives for the machine/deep learning (rather than the spiking neural network).

Table 1 Categories of different design options for hardware implementation of neuro-inspired computing. Representative porotypes are shown.

	Off-the-shelf technologies	CMOS ASIC	Emerging resistive synaptic devices
Level-based representation	GPUs [32] FPGAs [33]	TPU [34] CNN accelerators [12] [13, 14, 15]	Analog synapses: UCSB’s 12×12 crossbar array [37] Umich’s 32×32 crossbar array [38] Tsinghua’s 128×8 1T1R RRAM array [39] IBM’s 500×661 1T1R PCM array [40] UCSB’s 785×128 floating-gate transistor array [41]
			Binary synapses: ASU/Tsinghua’s 16 Mb 1T1R RRAM macro [41]
Spike representation	SpiNNaker [8]	Analog neuron: BrainScaleS [9].	IBM’s 256×256 1T1R PCM array with STDP neuron circuits [40]
		Digital neuron: TrueNorth [11]	

Now let us discuss why eNVM is attractive for the hardware implementation of neuro-inspired computing. To overcome the aforementioned challenges with the SRAM based synapses, the researchers are attracted by exploiting the unique properties of eNVMs to better serve the analog synapses in the neural network. The goal is to replace the SRAM arrays with the resistive crossbar arrays to store and/or update the weights in a more parallel fashion. Compared to the binary SRAM cell with 6 or 8 transistors, the eNVM cell occupies more than tens of times less area and can store multi-bit per cell, which further increases the integration density thereby supporting a larger capacity on-chip (for larger problem size or dataset). Storing most or all the weights on-chip thus eliminating the off-chip memory access is critical to the acceleration and the reduction of energy consumption from the entire system point of view. Thanks to the non-volatility, the eNVM devices can be powered off-and-on instantly and consume no standby leakage. In addition, unlike SRAM array’s sequential write and read, resistive crossbar array with eNVMs can do parallel programming and weighted sum for further speedup, potentially enabling the online training.

Generally speaking, eNVMs are mostly resistive memories that use “resistance” to represent and store data, although the ferroelectric memory uses “capacitance” to present and store data. The resistance based eNVMs including the spin-transfer torque magnetic random-access memory (STT-MRAM) [42], phase change memory (PCM) [20], resistive random access memory (RRAM) [21]. RRAM has two sub-categories: one is anion-based oxide random access memory (OxRAM) and the other one is cation-based conductive bridge random access memory (CBRAM) [43]. In some literature, resistive memories are also referred to as memristors [24]. In this review, we will focus on PCM and RRAM technologies as they have demonstrated the multilevel states, and also will briefly discuss the usage of the ferroelectric field-effect transistor (FeFET) and the floating-gate transistor (the basic cell for today’s flash memory technology) towards synaptic devices. The eNVMs are mostly pursued as the next-generation storage-class memory technologies with aggressive industrial research and development [22]. For example, Samsung has reported an 8 Gb PCM prototype chip in 20 nm node featuring 40 MB/s write bandwidth [44]. SanDisk/Toshiba has reported a 32 Gb RRAM prototype chip in 24 nm node [45]. Micron/Sony has reported a 16 Gb CBRAM prototype chip in 27 nm node featuring 200 MB/s write bandwidth and 1 GB/s read bandwidth. Panasonic has commercial products of micro-controllers with MB-capacity embedded TaO_x RRAM [46]. These demonstrations show that the eNVMs are viable technologies for the potential large-scale integration of the neural networks.

3. Device-level Characteristics of Synaptic Devices

3.1 Desirable characteristics

In this section, we will discuss the desirable characteristics for resistive synaptic devices for improving learning accuracy and energy-efficiency. Table 1 summarizes the desirable performance metrics for resistive synaptic devices. It should be noted that many of the metrics are highly application-dependent (related to different scenarios, e.g. online or offline training, and the dataset size, etc.).

Device Dimensions: The large-scale integration of neural networks requires a compact synaptic device with a small device footprint. Resistive synaptic devices with scalability down to sub-10 nm regime is preferred. Today’s RRAM and PCM devices have proven such scalability, however most of the demonstrations are for the digital memory application. Embedded floating-gate transistor (though a more mature technology) seems difficult to be scaled down to 28 nm or beyond. Ultimately, a two-terminal eNVM device (ideally with a two-terminal selector) that is compatible with the crossbar array architecture and three-dimensional integration is the target for research.

Multilevel States: Synaptic plasticity characteristics observed on biological synapses show an analog-like behavior with multilevel synaptic weight states. Most neuro-inspired algorithms also employ the analog synaptic weights to learn the patterns or extract features. In general, the more multilevel states (e.g. > hundreds of levels) could be translated into a better learning capability and an improved network robustness. However, the weight precision requirement (i.e. the number of conductance states) remains strongly application-dependent. Generally, for the online training requires more levels of states than the inference-only. We will have more in-depth discussion on precision reduction from the algorithm point of view in Section 5.2. If the multilevel states in the resistive synaptic devices are insufficient to meet the precision requirement, there are two alternative solutions: First, multiple devices could be grouped to represent higher precision at the expensive of area and energy [41]. Second, recent work shows that binary synaptic devices with stochastic weight update may equivalently provide the properties of analog synapses for some simple neural networks [47, 48].

Dynamic Range: Dynamic range means the on/off ratio between the maximum conductance and minimum conductance. Most of resistive synaptic device candidates exhibit a range of 2× to >100× range. The larger the dynamic range is, the better mapping capability of the weights in the algorithms to the conductance in the devices, because the weights in the algorithms are typically normalized within a range (e.g. between 0 and 1). Considering the power consumption for parallel reading the weights in a large-scale integration of neural networks (e.g. with a matrix size 512×512 or above), a guideline of the desired range of a single device could be from 1 nS to 100 nS.

Asymmetry and Linearity in Weight Update: The linearity in weight update refers to the linearity of the curve between the device conductance and the number of “identical” programming pulses. Ideally, this should be a linear and symmetric relationship for a direct mapping of the weights in the algorithms to the conductance in the devices. However, the realistic resistive synaptic devices generally have the nonlinearity in weight update. The trajectory of the weight increase (long-term potentiation, LTP) process differs from that of the weight decrease (long-term depression, LTD) process, resulting in the asymmetry as well. The conductance tends to change rapidly at the beginning but saturate at the end of the processes. Figure 2 (a) shows an example of the TaO_x/TiO_2 device conductance under identical programming pulses [49, 50]. This nonlinearity/asymmetry is undesired because the change of the weight (ΔW) depends on the current weight (W), or in other words, the weight update has a history dependence. Recent results have shown that this nonlinearity/asymmetry has caused the learning accuracy loss in the neural networks [51, 49]. We will have more in-depth discussion on the impact of weight update nonlinearity in Section 5.2. There are a few strategies to improve the linearity by optimizing the programming schemes. For example, identical pulse pairs (a larger pulse followed by a smaller pulse with reversed polarity) could improve the nonlinearity of the TaO_x/TiO_2 device, as shown in Figure 2 (b). Non-identical pulses with varying widths could further improve the nonlinearity of the TaO_x/TiO_2 device, as shown in Figure 2 (c). However, the non-identical pulse generation requires non-trivial design efforts from the peripheral circuit’s perspective. Therefore, in this review, we only consider the case when identical pulses are used to update the weights. It should be noted that the weight update nonlinearity/asymmetry is a key issue only for online training, which requires a smooth and continuous conductance tuning, while for offline training, the nonlinearity could be shadowed by the iterative programming with write-verify technique (see discussions in Section 3.2 B).

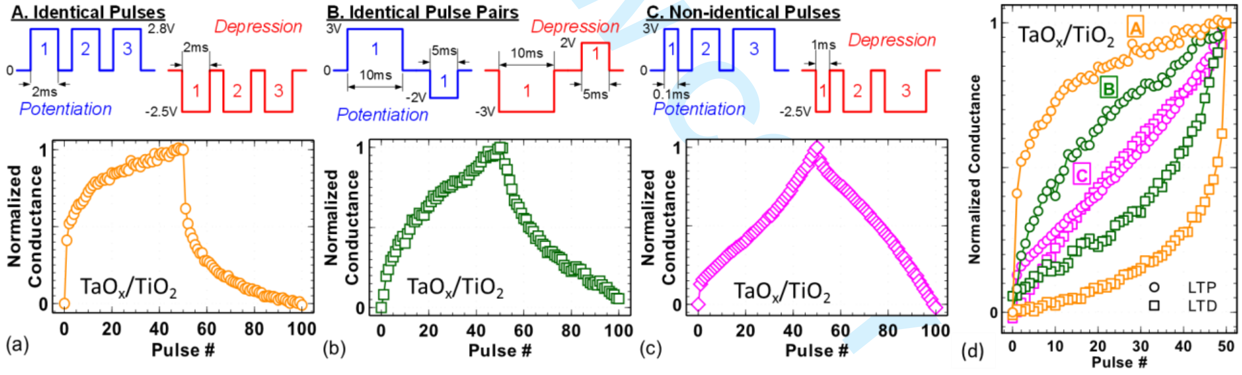


Figure 2 The weight update behavior (conductance vs. # pulse) of TaO_x/TiO_2 device with different pulse schemes. Non-identical pulses with varying pulse widths could improve the nonlinearity/asymmetry, but complicates the peripheral circuitry design. Adapted from [49].

Programming Energy Consumption: The estimated energy consumption per synaptic event is around 1~10 fJ in biological synapses. Most RRAM devices show a programming energy around 100 fJ~10 pJ, while most PCM devices may have even higher programming energy 10~100 pJ. The fundamental challenge is that it is much more difficult (thus paying more energy) to move the ions/defects in solid-state devices than moving calcium ions in the liquid environment in biological synapses. A back-of-envelop calculation is given as follows. In biological synapses, the spike voltage is ~10 mV, the ionic current ~1 nA, the spike period ~ 1 ms, therefore the energy is about 10 fJ. In resistive synaptic devices, the typical programming voltage is ~1 V, the programming current is typically ~µA, although the programming speed can be accelerated less than the real-time to be ~µs, still the energy is on the order of pJ. Further device engineering is thus needed to reduce the energy consumption by improving the programming speed down to ~ns regime.

Retention and Endurance: During the online training, the weights are frequently updated, and the data retention requirement can be relaxed. When the training is complete, the resistive synaptic devices should behave as a long-term memory with a data retention in the order of 10 years at the maximum chip operating temperature (e.g. 85 °C). The number of cycling endurance is much application-dependent, relying on how many weight updates are required in the training processes. For a relatively simple task (i.e. the MNIST handwritten digit recognition [52]), 60,000 training images with 50 training epochs (to be repeated) gives a maximum weight update possibility to be 3×10^6 updates. Actually not every synapse is updated in the training in each cycle, thus an endurance ~10^4 cycles is sufficient for training MNIST dataset [41]. However, considering more challenging tasks (i.e. ImageNet Challenge [5]), much more endurance may be required. It should be pointed out that the definition of the endurance cycles is tricky in the resistive synaptic devices, because each weight update is generally a small incremental change in the analog conductance tuning, thus it is unlike the full switching from the on-state to the off-state in a binary eNVM.

Uniformity and Variability: Poor uniformity or significant variability in eNVMs is a major barrier for digital memory applications. In contrast, the neural networks promise a potential robustness against device variations. The device variations could partially be tolerated by two mechanisms: the massive (thus maybe redundant) connections between neuron nodes by synaptic arrays, and the iterative weight update process during the online training. The degree of variations that can be tolerated at the system-level strongly depends on the network architecture and the accuracy required by the target application. Recent results have shown the reasonable robustness against device variations in different neural networks [49, 53]. However, for offline training (with write-verify), the requirement on the uniformity is more stringent because the network could not adapt itself for inference-only. We will have more in-depth discussion on the impact of variations in Section 5.2.

Table 2 Summary of the desirable performance metrics for synaptic devices.

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

Note: * these numbers are application-dependent

3.2 Representative materials systems and device prototypes

In the past few years, many resistive synaptic device candidates that are capable of tens to hundreds levels of conductance states have been demonstrated at single device level. In addition to the analog conductance tuning capability, biologically realistic behaviors such as short-term memory, pair-pulse facilitation, and spike-timing-dependent plasticity have been emulated in various devices including Ag/Ag_2S [54], Cu/Cu_2S [55], Ag/GeS_2 [56], $Ag/Ge_{30}Se_{70}$ [57] Ag/SiO_xN_y [58] based CBRAM, TiO_x [59], HfO_x [60, 53, 61], WO_x [62, 63] and TaO_x [64] based OxRAM, etc. However, how these bio-plausible features could facilitate the computation at the system-level is unclear so far, thus in this review, we will only survey the analog weight update characteristics of the reported devices.

A. PCM

The resistance change in PCM relies on the reversible crystallization and amorphization of the chalcogenide materials [20], typically $Ge_2Sb_2Te_5$. The crystalline phase has a lower resistance (higher conductance) than the amorphous phase, and multilevel resistance states could be achieved by controlling the volume of the amorphous region. The larger ratio of the amorphous volume over the crystalline volume will result in a larger resistance. Therefore, the PCM device could behave as an analog synapse. The realization of PCM based synaptic devices could be dated back to the work [65], where the device conductance could be gradually increased or decreased with ~100 states by applying a sequence of programming pulses with increasing amplitudes. The STDP learning rule has also been demonstrated by designing the appropriate pulse waveforms. Then various pulse programming schemes have been proposed by different groups to reduce the complexity and power consumption of neuromorphic circuits using the PCM devices [66, 67, 68, 69]. One challenge of PCM based synaptic device is the relatively more abrupt RESET (weight decrease) process than the SET (weight increase) process. This is because the melting and quench in the RESET is less controllable than the partial crystallization in the SET. Figure 3 (a) shows multilevel states are achieved by identical SET programming pulses, while Figure 3 (b) shows only binary states are achieved by identical RESET programming pulses. To address this challenge, a design of 2-PCM synapse has been proposed [70]: one is used to implement synaptic potentiation (LTP-device) while the other one is used to implement synaptic depression (LTD-device). In both cases, the device undergoes partial crystallization (i.e. gradual SET) process. With this scheme, the conductance of both PCM devices keeps increasing when undergoing LTP and LTD, the contribution of the currents through the LTP device is positive while the contribution through the LTD device is negative in the differential output stage. The negative current through the LTD device acts like synaptic-depression, because the current flowing through it is subtracted in the differential output stage. The operation principle of the 2-PCM synapse device is shown in Figure 3 (c).

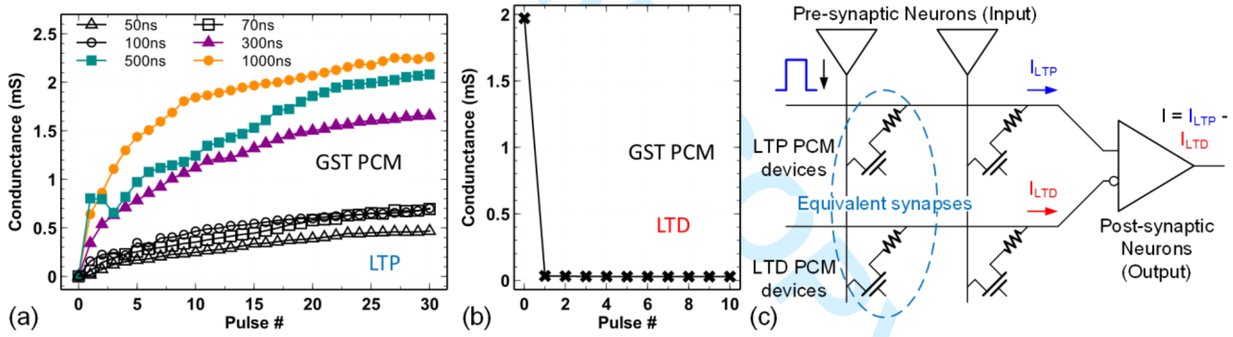


Figure 3 GST based PCM weight update: (a) Weight increase (LTP) (b) Weight decrease (LTD). Gradual LTD is more difficult to achieve in PCM due to the abrupt RESET process. (c) Schematic of the concept of 2-PCM synapse to avoid the abrupt RESET process. The contribution of the current from the LTD device is subtracted at the post synaptic neuron. Adapted from [70].

B. RRAM

Generally, there are two kinds of switching mechanism of RRAM devices, one is based on the filamentary mechanism, in which the conductive filaments with metal ions or oxygen vacancies form and rupture in the insulating layer; the other type is based on the interfacial mechanism, in which the distribution of oxygen vacancies at the interface (e.g. oxide/oxide interface, or electrode/oxide interface) is modulated by the electric field. For the conventional memory application, the filamentary RRAM is widely adopted. However, the filamentary RRAM has typically shows an abrupt SET (weight increase) process, and a gradual RESET (weight decrease) process. Note this trend is just opposite to the typical PCM device. Thus in the early design of RRAM based synapse, one-way RESET-only learning scheme was used in HfO_x based synaptic device [71]. The abrupt SET is attributed to the positive feedback between the filament growth speed and the electric field, resulting in the formation of a single dominate strong filament [72]. To make the SET process gradual, one way is the oxide stack engineering (i.e. bilayer oxides) to make weak or multiple weak filaments as demonstrated in TaO_x/HfO_2 [73] and AlO_x/HfO_2 [74] devices. On the other hand, the interfacial device typically shows both gradual SET and RESET processes, as demonstrated in Ag:a-Si [75], $PrCaMnO_3$ (PCMO) [76, 77] and TaO_x/TiO_2 devices [78, 79].

Depending on the application scenarios, for online training or for offline training, different programming schemes could be used thus the requirement on device characteristics may be different. For instance, for offline training, the write-verify technique could be used to iteratively program the conductance states to the pre-defined target level, since it is a one-time programming process and the programming speed is not a priority but the programming accuracy is. Typically, a pulse sequence (programming-read-programming-read …) is applied as shown in Figure 4 (a) [80]. The higher amplitude programing voltage pulses could be used to reach the desired resistive state faster but also at a cruder precision. On the other hand, smaller amplitude pulses will approach the state at a finer step but may require an exponentially longer time. It is therefore natural to use a variable amplitude pulse sequence to approach the desired state in optimal time. With no variations in switching dynamics, this could be achieved by applying a sequence of decreasing amplitude voltage pulses with every new pulse driving the device closer to the desired state (Figure 4 (b)). Because of device-to-device variations calculating the parameters of the initial pulse is challenging. Somewhat counterintuitively, one possible solution is to use sequences of increasing amplitude voltage pulses (Figure 4 (c)) instead, which always starts with small non-disturbing pulse. The device conductance is checked by applying read pulse after each write pulse. Such alternating read/write sequence is applied until either the desired tuning accuracy is reached or overshooting occurs. In the latter case, the new sequence of opposite polarity is started. Because this time the initial state would be typically closer to the desired one, the final maximum amplitude of the write voltage pulse in that new sequence will be smaller as compared to that of earlier sequence, which in turn ensures driving the device closer to the desired state.

For single $Pt/TiO_{2-x}/Pt$ devices, the algorithm allows tuning the conductance with 1% precision (which is equivalent to ~ 8 bit) to any desired value within device's dynamic range, as shown in Figure 4 (d) [80]. Due to half select problem the accuracy is expected to be lower, e.g. about 3% as demonstrated in small crossbar circuits. For Ag/a-Si/Pt single devices, the tuning accuracy is also close to 1% for low resistive states [81]. It should be noted that one of the factor limiting accuracy for high resistance states in these devices is intrinsic random telegraph noise (RTN). Similar iterative programming schemes were demonstrated in HfO_x devices as well [82].

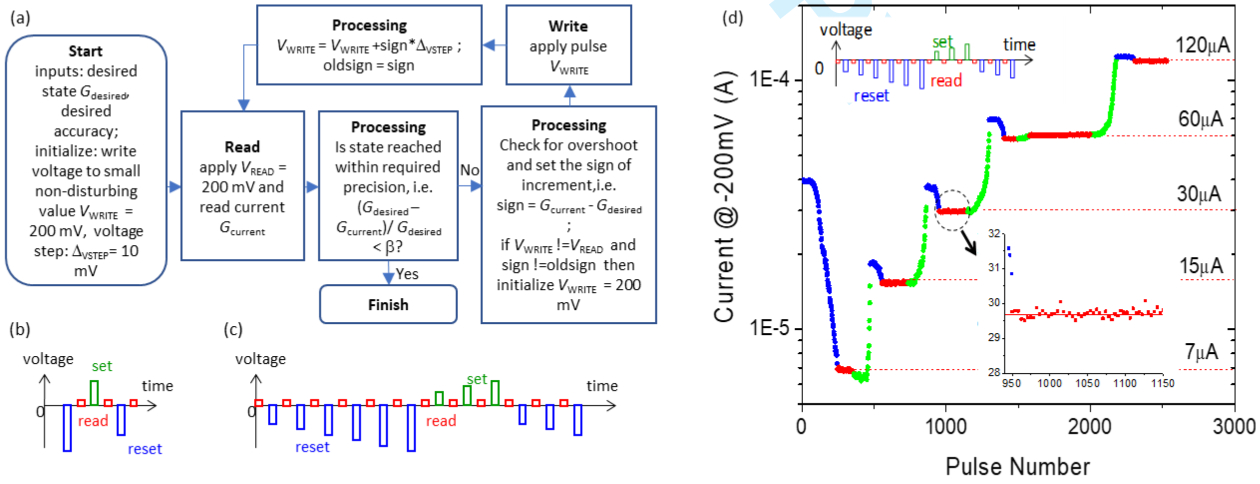

(a)
Start
inputs: desired state G_desired, desired accuracy; initialize: write voltage to small non-disturbing value V_WRITE = 200 mV, voltage step: Δ_VSTEP= 10 mV
Read
apply V_READ = 200 mV and read current G_current
Processing
Is state reached within required precision, i.e. (G_desired – G_current)/ G_desired < β?
No
Yes
Finish
Processing
Check for overshoot and set the sign of increment,i.e. sign = G_current - G_desired ;
if V_WRITE !=V_READ and sign !=oldsign then initialize V_WRITE = 200 mV
Processing
V_WRITE = V_WRITE +sign*Δ_VSTEP ; oldsign = sign
Write
apply pulse V_WRITE
(b)
voltage
set
time
0
read
reset
(c)
voltage
set
time
0
read
reset
(d)
voltage
set
time
0
read
reset
Current @ -200mV (A)
1E-4
1E-5
120μA
60μA
30μA
15μA
7μA
32
31
30
29
28
950
1000
1050
1100
1150
0
1000
2000
3000
Pulse Number

Figure 4 Variation tolerant high precision tuning algorithm for offline training: (a) algorithm block diagram, (b) intuitive and (c) actually implemented voltage pulse sequence for tuning to a desired conductance state. High precision (~1%) tuning of analog memory demonstration in $Pt/TiO_{2-x}/Pt$ devices. Adapted from [80].

On the other hand, for online training, a smooth conductance tuning without write-verify is preferred, as the weights are trained on-the-fly, and the programming speed does not allow the back-and-forth iterative programming. Figure 5 shows some examples of state-of-the-art RRAM based devices in the literature that exhibit the bidirectional gradual conductance tuning under programming voltage pulses. Tens or even hundreds of multilevel states have been demonstrated, however, the weight update nonlinearity and asymmetry commonly exist. Note that the weight decrease curves in Figure 5 are mirrored back (as opposed to that in Figure 2).

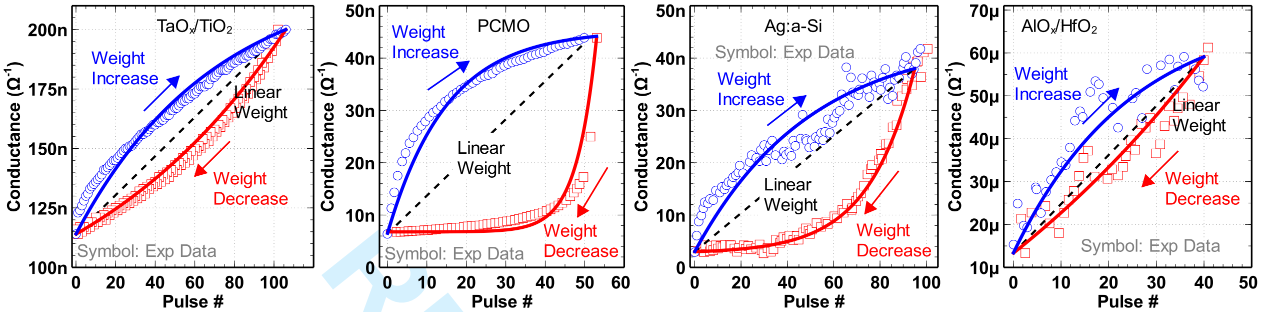


Figure 5 Representative analog synaptic device weight update behaviors from the literature: TiO$_x$/TiO$_2$ [78], PCMO [77], Ag:a-Si [75], AlO$_x$/HfO$_2$ [74].

C. FeFET

The ferroelectric field-effect-transistor (FeFET) synaptic device is a three-terminal structure that decouples the weight tuning and weight read path: the weight tuning relies on the programming voltage applied to the gate, while the weight current is read out by the drain-to-source current. Due to the three-terminal nature, FeFET is organized to a pseudo-crossbar array architecture for weighted sum (see more discussions in Section 5.1 A on pseudo-crossbar array). The physical mechanism of FeFET utilizes the multi-domain effects present in ferroelectric materials (i.e. the doped HfO$_2$) to gradually tune the gate capacitance and consequently the threshold voltage (V$_T$) and the channel conductance by the application of short voltage pulses to the gate [83, 84]. A recent experimental demonstration of analog FeFET synaptic device used the gate last fabrication process flow of an n-channel FeFETs [85], as shown in Figure 6. The gate stack consists of 10 nm Hf$_{0.5}$Zr$_{0.5}$O$_2$ (HZO) deposited by atomic-layer deposition n p-Si with 0.8 nm interfacial SiO$_2$ layer, and 600 ℃ anneal gives rise to multiple ferroelectric domains within a nanocrystaline structure of the HZO (see the inset figure of the transmission electron microscopy). The synaptic behavior in response to pulse Schemes 1-3 are shown in Figure 6 (a)-(c). Similar as in Figure 2, non-identical pulses with increasing widths or amplitudes improve the nonlinearity of the weight update curve, and Scheme 3 (with 75 ns width) exhibits the largest number of states, 32 (5-bit) and an on/off ratio of 47×. Compared to previous reported RRAM based analog synaptic devices, FeFET shows some promising features such as enlarged on/off ratio and shorter programming pulse width, as well as less variations in the weight update curve.

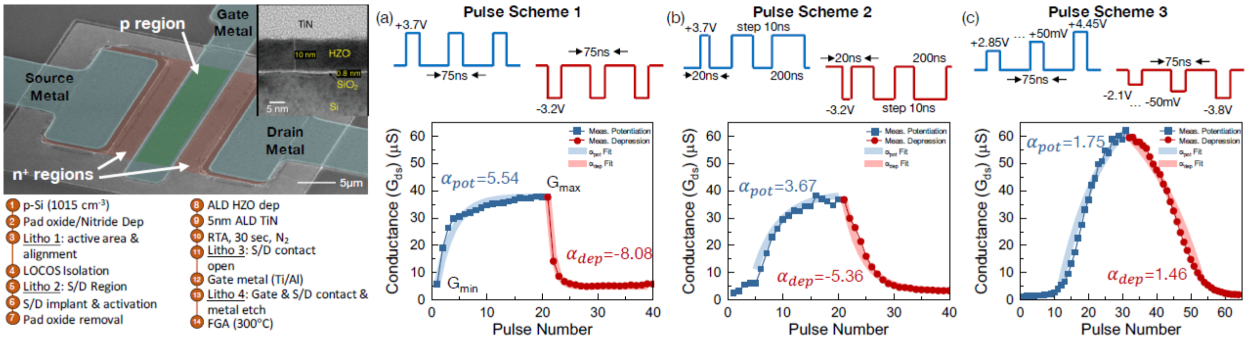


Figure 6 The fabrication process of doped HfO_2 based FeFET. Pulse schemes and corresponding weight update curves for (a) Pulse scheme 1 with identical pulses, (b) Pulse scheme 2 with increasing pulse widths and (c) Pulse scheme 3 with increasing pulse amplitudes. Adapted from [85].

4. Array-level Demonstration of Crossbar Array for Dot-Product Acceleration

4.1 The principle of weighted sum and weight update in crossbar array

The resistive crossbar array architecture has been proposed for implementing the weighted sum (or matrix-vector multiplication, dot product operation) [86], which is the mostly time-consuming step in the neuro-inspired learning algorithms. As shown in Figure 7, the crossbar array consists of perpendicular rows and columns with the resistive synaptic devices sandwiched at each cross-point. The weights in the neural network are then mapped to the conductance of the resistive synaptic devices.

The weighted sum operation is performed in a parallel fashion: read voltages are applied to all the rows, and then the read voltages are multiplied by the conductance of the synaptic devices at each cross-point, resulting in a weighted sum current in each column. Typically, the neuron circuits are placed at the end of the column to convert this analog current to the digital output or spikes. The proposed crossbar array architecture only performs the analog computation in the array core, and the communication between arrays is still through digital fashion considering the signal integrity issues in the on-chip routing channels. Although the input vectors could be represented by the analog voltage, it is better to be represented by the digital number of pulses. This is because the I-V nonlinearity of the resistive synaptic device may distort the weighted sum accuracy if using analog voltage ,and it is also difficult to generate multiple bias levels within a small read voltage range from peripheral circuit design’s perspective [87]. It is also worth pointing out that the sneak path problem for the conventional crossbar memory does not exist here if all the rows and columns are activated during the weighted sum. This is because the conventional memory requires reading out data by bit or by row, while all the cells here participate in the computation essentially following the Kirchhoff Law. The IR drop problem along long interconnect wires still exists here for a large-scale array, as the interconnect resistance may distort the weighted sum accuracy if it is a significant portion of the synaptic device resistance. The interconnect resistance effect can be mitigated by either relaxing the wire width [87] or re-mapping the weights of the weights in the algorithms to the conductance in the devices [88].

The weight update operation can be performed row by row (or column by column) in the crossbar array. In this case, selectors with threshold switching I-V (e.g. FAST selector [89]) may be needed to minimize the leakage current in other unselected rows/columns. In principle, the weight update can be performed in a fully parallel fashion on the entire array as the programming voltages can be applied from both ends (row and column) to the synaptic device [78, 90]. However, to program the entire array simultaneously usually demands a huge instant power from the peripheral circuits which seems not very feasible in practical designs.

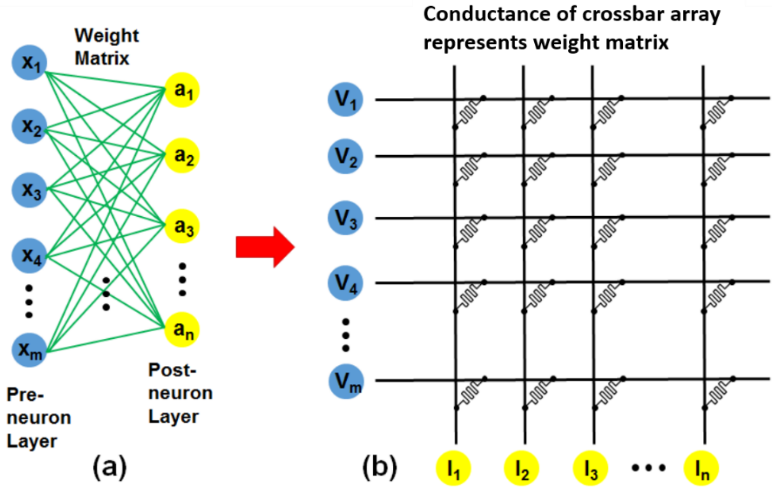


Figure 7 (a) The weight matrix between neuron layers in the neural network. (b) The crossbar array consists of perpendicular rows and columns with the resistive synaptic devices sandwiched at each cross-point. The weights in the neural network are mapped to the conductance of the resistive synaptic devices. The weighted sum operation is performed by applying read voltages to all the rows and read out the weighted sum current in all the columns.

4.2 Array-level experimental demonstration

Although various resistive synaptic device prototypes have been reported in the literature, most of these work still focus on the single device characterization. The early design exploration of array-level performance is based on simulation only with the device model fitted with single device measurement data, as pioneered in [91, 56].

Recently, there have been a few experimental demonstrations of simple neural network on small-scale to medium-scale arrays. For example, K.-H. Kim, et al. [92] demonstrated one-time programming weights into 40×40 Ag: a-Si crossbar array. S. B. Eryilmaz, et al. [93] employed a Hopfield network consisting of a 10×10 PCM 1T1R array and 10 recurrently connected software neurons for the implementation of associative learning. Later, using the same platform with 2-PCM per synapse, S. B. Eryilmaz, et al. [94] demonstrated a Restricted Boltzmann Machine (RBM) with 9×5 synapses, a generative probabilistic graphical model as a key component for unsupervised learning in deep neural network. S. Park, et al. [95] demonstrated a single-layer perceptron network in 32×6 PCMO crossbar array with off-chip neuron circuits on a printed circuit board. In this experiment, the human thought patterns corresponding to three vowels, i.e. /a /, /i /, and /u/, were in-situ learned and recognized using electroencephalography (EEG) signals generated while a subject imagines speaking vowels. P. M. Sheridan, et al. [38] demonstrated a 32×32 WO_x crossbar array with offline trained analog weights for implementing the sparse coding algorithm for unsupervised feature extraction. M. Hu, et al. [96] developed a 64×64 TiO_x based 1T1R array with 6-bit (64 levels) offline training and a single-layer perceptron for MNIST image recognition. L. Gao, et al. [97] proposed a scheme to implement the convolution kernel in CNN by unrolling 2D kernel matrix into 1D column vector and demonstrated the concept in a 12×12 HfO_x crossbar array.

Furthermore, with more mature PCM and floating-gate transistor technologies, a few full functional macro chips have been developed as well. For example, S. Kim, et al. [40] demonstrated a 64 kb (256×256) PCM 1T1R array with on-chip leaky integration and fire neuron circuits for continuous in-situ STDP learning. J. Lu, et al. [98] developed a machine learning prototype chip in 130 nm node with floating-gate synapses, exhibiting a remarkable energy efficiency 480 GOPS/W in the training mode and 1 TOPS/W in the inference mode.

The aforementioned demonstrations show the promises for future large-scale integration with eNVMs for neuro-inspired computing. In the next, we will present a few representative array-level demonstrations that perform the online training with the eNVMs inside the array. However, it should be pointed out that the weighted sum is still performed row-by-row sequentially in demonstrations in Section 4.2 A and B due to the design limitations in the 1T1R array to turn on one row at a time.

A. IBM's 500×661 1T1R PCM array for MNIST recognition

Using 2-PCM per synapse, G. W. Burr, et al. [51] demonstrated a 3-layer perceptron (fully-connected ANN) with 164,885 synapses, trained with backpropagation on a subset (5,000 examples) of the MNIST database of handwritten digits, as shown in Figure 8 (a). The experiments were done using software based neurons which perform the nonlinear activation function, while the weighted sum and weight update were measured and implemented with the 500×661 1T1R PCM array. The weight update in backpropagation was done using a modified delta rule that sends the stochastic pulses from rows and columns, and the overlap of the two pulses becomes the effective programming time window. It is proved that this weight-update modification does not degrade the classification accuracy in the testing dataset as compared to the case when the network was trained in software. However, nonlinearity and asymmetry in PCM's weight update limit the learning accuracy in this hybrid hardware-software experiments to 82-83%, as shown in Figure 8 (b), though the learning accuracy could achieve ~97% in the software baseline. Asymmetry (between the gentle conductance increases of PCM partial-SET and the abrupt conductance decrease of a PCM RESET operation) was mitigated by an occasional RESET strategy, which could be both infrequent and inaccurate. While in these initial experiments, network parameters such as learning rate η had to be tuned very carefully, a modified "Local Gains" algorithm offered wider tolerance to η, higher classification accuracies, and lower training energy as proposed in this group's later work [18]. The sensitivity analysis [99] showed that eNVM-based ANN can be expected to be highly resilient to random effects (e.g. variability, yield, and stochasticity), but highly sensitive to "gradient" effects that act to steer all synaptic weights. It is shown that an ideal bidirectional eNVM with a symmetric, linear weight update of finite but large dynamic range can deliver the same high classification accuracy on the MNIST dataset as a conventional, software-based implementation.

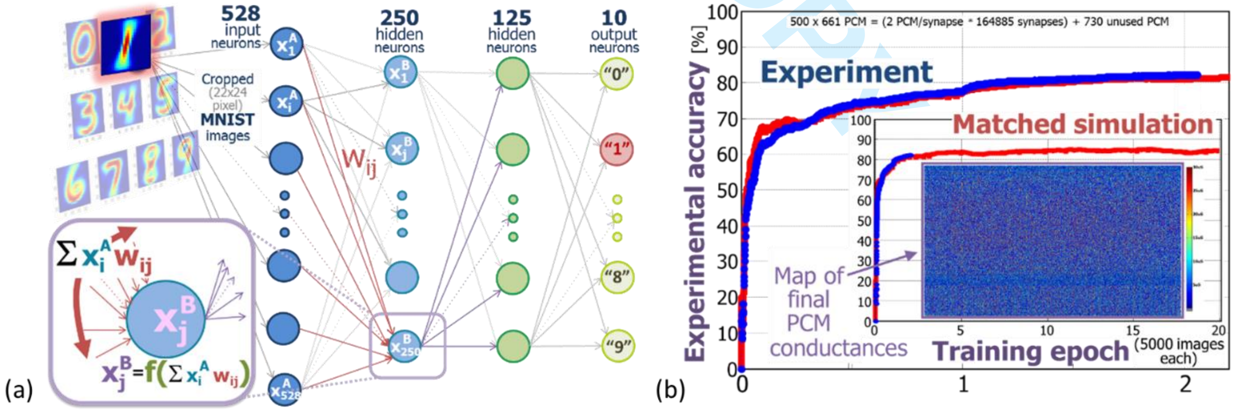


Figure 8 (a) The implementation of 3-layer perceptron with PCM synapses. In feed forward propagation, each layer's neurons drive the next layer through weights w_{ij} and a nonlinear neuron activation function *f*(). Input neurons are driven by input (for instance, pixels from successive MNIST images (cropped to 22×24)); the 10 output neurons classify which digit was presented. (b) Learning accuracy for a 3-layer perceptron of 164,885 synapses with 2-PCM per synapse, with all weight operations taking place on a 500×661 PCM 1T1R array. Also shown is a matched computer simulation of this network, using parameters extracted from the experiment. Adapted from [51].

B. Tsinghua's 128×8 1T1R analog RRAM array for face recognition

As shown in Figure 9, P. Yao, et al. [39] demonstrated a 1-layer perceptron for face recognition with 128×8 1T1R analog RRAM array, as shown in the micrograph of the array. Different than the unidirectional conductance tuning in PCM as used in Section 4.2 A, bidirectional analog conductance modulation was achieved in $TaO_x/HfAl_yO_x$ RRAM stack, which was integrated on top of a CMOS transistor to form the 1T1R structure. The network was trained online to recognize and classify grey-scale face images from the Yale Face Database [100] and tested with the extra unseen faces as well as constructed images with noisy pixels. The experiments include two phases: inference and weight update. As for the inference phase, the 9 training patterns (belong to three people) were input to the network on bitline (BL) side as read voltages. These 9 patterns were chosen from the Yale Face Database and sequentially cropped and down-sampled to 320 pixels in 20×16 size. The total currents measured on source line (SL) side (3 lines) is applied to a nonlinear activation function in software neuron to predict 3 classes of faces. In the weight update, two update rules were proposed, one is without write-verify (1) which only points out the switching direction depending on the error's sign, following the Manhattan rule; and the other one is with write-verify (2) which implements both direction and amplitude based on the error's sign and value, following the delta rule. Without write-verify scheme simplifies the control circuitry since it is not necessary to calculate the specific analog value of the error between the targeted weight and the current weight, but may slow down the converging speed due to the programming stochasticity. There is a trade-off between these two schemes: with and without verify schemes could achieve a relatively high recognition rate after converging, i.e. 91.7% and 87.5% respectively. The scheme with verify shows 4.61X faster converging speed, 1.05X higher recognizing accuracy, and 4.41X lower energy consumption. The network was also tested with random noises in the image pixels and can maintain its classification accuracy up to 31.25% noise.

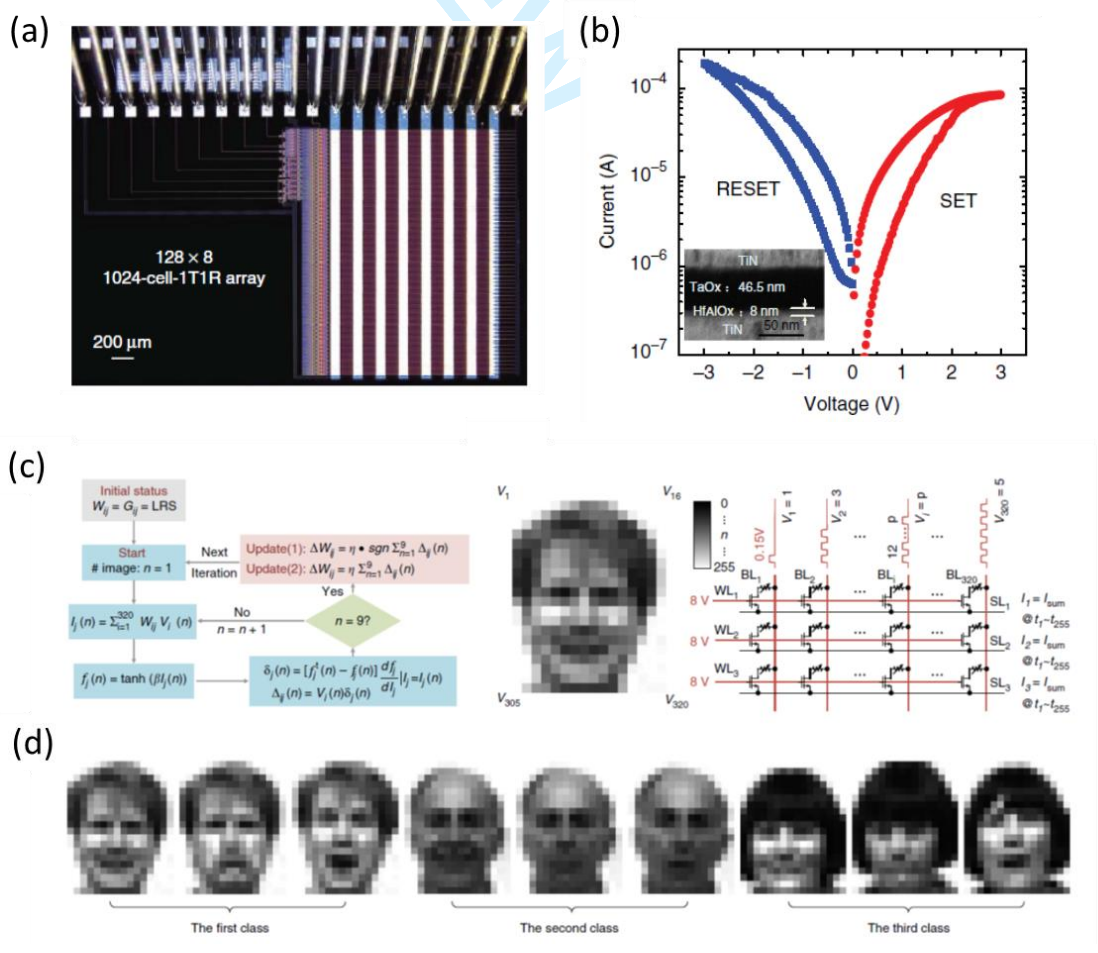


Figure 9 (a) The micrograph of the fabricated 128×8 1T1R array using fully CMOS compatible fabrication process. (b) The RRAM device stack is based on TaO_x/$HfAl_yO_x$ and integrated on top of a CMOS transistor, with a bidirectional gradual I-V characteristic for analog conductance tuning. (c) The flowchart of the 1-layer perceptron model for the training process and two weight update rules were proposed, one is without write-verify (1) which only points out the switching direction; and the other one is with write-verify (2) which implements both direction and amplitude. The schematic of fully parallel read operation and how a pattern is mapped to the input. (d) The 9 training patterns belongs to three classes, a subset of Yale Face Database. Adapted from [39].

C. UCSB's 12×12 crossbar array for pattern recognition

Unlike the 1T1R array used above, the practical implementation of memristor-based neural networks in a true crossbar array, even of their simplest variety such as multilayer perceptron (MLP) network, is still challenging, mainly due to its immature fabrication technology. The most critical requirement to the technology is to ensure a relatively low (within one octave) distribution of device forming and switching threshold voltages. This condition enables individual forming, and then fine-tuning of every memristor of the crossbar, without disturbing already formed devices. Memristors featuring low variability bilayer Al_2O_3/TiO_{2-x} were recently reported in [37, 101]. The optimized technology was then used for the fabrication of integrated 12×12 crossbars, as shown in Figure 10 (a)-(b). The crossbars featured a high uniformity of virgin (pre-formed) and post-formed in the switching voltages, as shown in Figure 10 (c)-(d).

The fabricated memristive crossbar was used to implement a simple neural network (a single-layer perceptron) with 10 inputs and 3 outputs, fully connected with 30 synaptic weight, as shown in Figure 10 (e). Such network is sufficient for performing, for example, the classification of 3×3-pixel black-and-white images with 9 network inputs ($V_1,\ldots,V_9$) corresponding to pixel values, into 3 classes (Figure 10 (f)). One more input, V_{10}, was used for the source of 3 adjustable biases of nonlinear activation functions. The network was tested on a set of 30 patterns including 3 stylized letters ("z", "v", and "n") and 3 sets of 9 noisy versions of each letter, formed by flipping one of the pixels of the original image - see the inset in Figure 10 (g). Because of the limited set size, it was used for both training and testing.

Physically, each input signal was represented by voltage V_j equal to either +0.1 V or -0.1 V, corresponding, respectively, to the black or the white pixel. The bias input V_{10} was -0.1V. Each synaptic weight was implemented with a pair of memristors, so that $w_{ij} = G_{ij}^{+} - G_{ij}^{-}$, enabling negative weights values. The effective conductances $G_{ij}^{\pm}$ were in the range from 10 to 100 µS, so that the output currents I_i were of the order of a few µA. The network was trained "in situ", i.e. without using its external computer model, with the so-called Manhattan Update Rule, which is essentially a coarse-grain, batch-mode variation of the usual Delta Rule of supervised training. At each iteration ("epoch") of the procedure, the training set patterns were applied, one by one, to network's input, and its outputs $f_i(n)$, where n is pattern's number, were used to calculate the weight increments. Once all patterns of the training set had been applied, and all due increments ΔG were calculated, and the synaptic weights modified.

In the demonstrated system, the weights were modified in parallel for each half-column of the crossbar (corresponding to a certain value of index i in the above formulas), using two sequential voltage pulses. Namely, first a "set" pulse with amplitude V_{w+} = 1.3 V was applied to increase conductances of the synapses whose ΔG had been positive; then a "reset" pulse V_{W-} = -1.3V was applied to the remaining synapses of that half-column. This fixed-amplitude pulse procedure followed the Manhattan Update Rule only approximately, because the actual increment of conductance G depends on its initial value. Due to this specific (though quite representative) switching dynamics, the best classification performance was achieved when the memristors had been initialized somewhere in the middle of their conductance range, around 35 µS. At such initialization, the perfect classification was always reached - on the average, after 23 training epochs – see Figure 10 (g).

The main advantage of memristors is their very low chip footprint, determined only by the overlap area of the metallic electrodes. Because of that, many types of RRAM (or memristor) may be scaled down below 10 nm without sacrificing their endurance, retention, and tuning accuracy, with some of the properties (such as the on/off conductance ratio) being actually improved [102]. Moreover, these devices are naturally suitable for 3D integration – see, e.g. recent simple demonstrations of such an integration in Figure 11 [103, 104]. On the other hand, the tuning of memristors is based on reversible displacements of just a few atoms, so that even the best technologies of their fabrication developed by now do not yet provide the device variability low enough for VLSI circuits. The main hope is that the memristor neural networks may be able to piggyback on the intensive effort by several major industrial chipmakers toward the development of the technology of these devices for ultra-dense 2D and 3D NVMs.

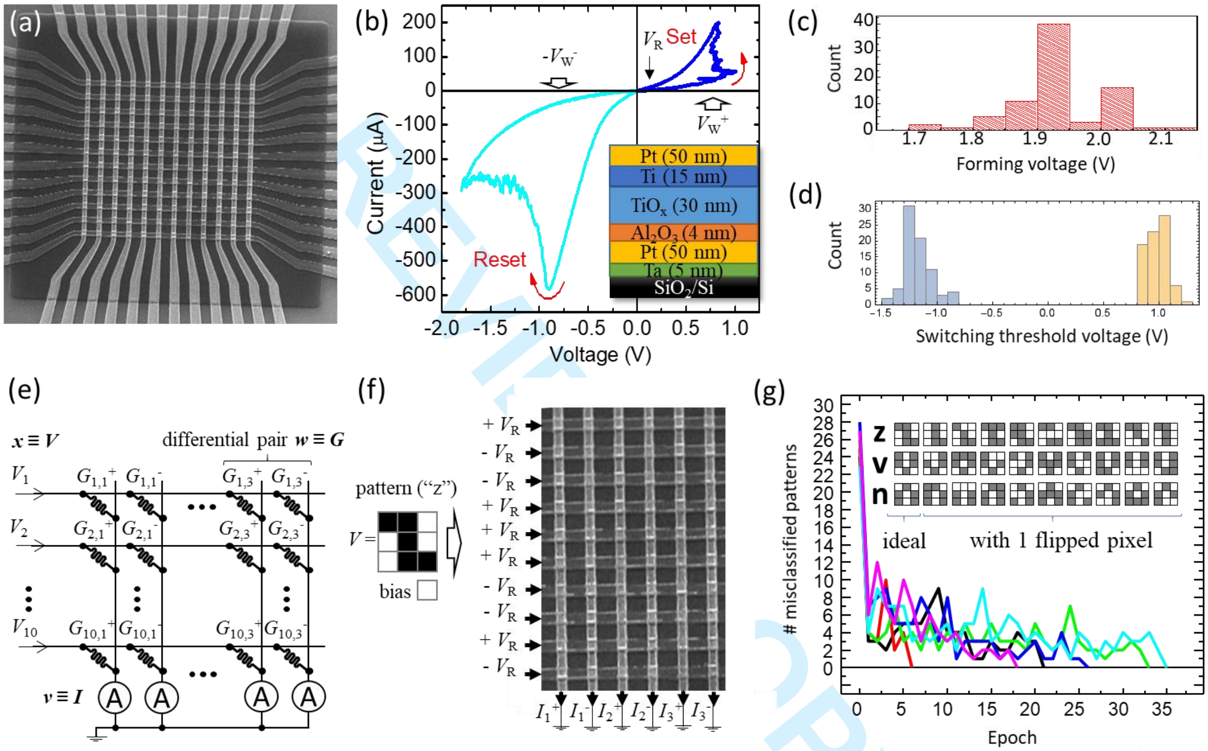


Figure 10 Perceptron classifier demonstration: (a) integrated 12×12 crossbar with an Al_2O_3/TiO_{2-x} memristor at each cross-point; (b) a typical *I*-*V* curve of a formed memristor; histograms of forming voltages (c) and effective switching thresholds voltages (d) for set and reset transitions; (e) perceptron implementation using a 10×6 fragment of the memristive crossbar; (f) example of the classification operation for a specific input pattern; and (g) the convergence of network outputs, in the process of training, to the perfect (zero-error) set, for 6 different initial states. The classification was considered successful when the output signal corresponding to the correct class of the applied pattern was larger than all other outputs. The insets in panels (b) and (g) show device's cross-section and the used input pattern set, correspondingly. On panel (d), the positive / negative switching threshold voltages were defined as the smallest amplitudes of 500-µs voltage pulses that caused resistance change by more than 2 kΩ in memristors pre-set to their high / low resistive states. Adapted from [37, 101].

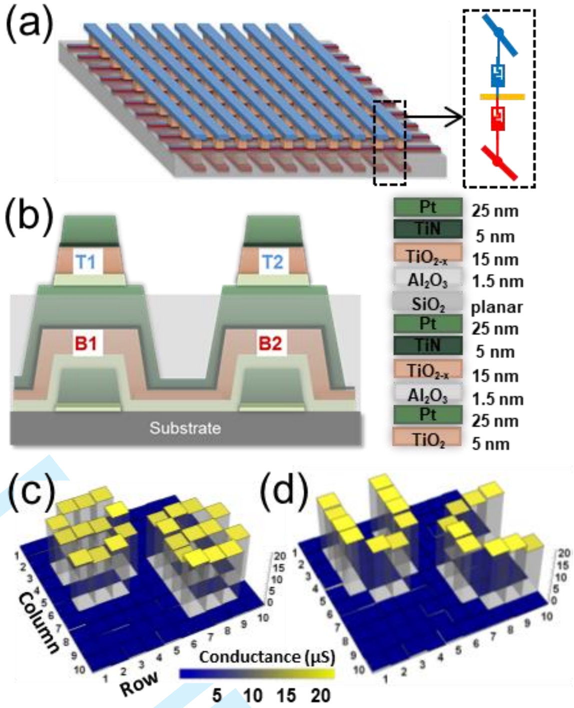


Figure 11 3D integration of memristor crossbar: (a) Circuit, (b) cross-section, and (c, d) experimental results of two vertically integrated TiO_x planar memristor crossbars. Adapted from Ref. [103].

D. UCSB's floating-gate array for MNIST image recognition

Another unique opportunity is offered by the floating-gate memory technology, which can now be embedded in CMOS logic process. Such cells are naturally larger as compared to 2-terminal memristor (though quite comparable with its 1T1R version). However, their main advantage is very mature fabrication technology. Custom design has been recently demonstrated for the industrial-grade 180-nm [105, 106] and 55-nm [107] (Figure 12 (a)) NOR flash memories. Naturally, floating-gate cells are suitable as adjustable conductances in a pseudo-crossbar fashion, with accuracy better than 1% (Figure 12 (b)-(c)), provided that the memory blocks are modified to allow for individual, precise adjustment of the conductance of each device. Such modification has already enabled a successful implementation of a medium-scale (28×28-binary-input, 10-output, 3-layer, 101,780-synapse) network for MNIST image classification (Figure 12 (d)-(e)) [108]. Remarkably for such a first attempt, still using the older 180-nm technology, the experimentally measured time delay and energy dissipation (per one pattern classification) were below, respectively, 1 μs and 20 nJ, i.e. at least three orders of magnitude better than the 28-nm TrueNorth chip implementation of the same task [109], with a similar accuracy. The experimental results for the chip-to-chip statistics, long-term drift, and temperature sensitivity show no evident showstoppers for the practical deployment of such networks. The estimates [108], based on the experimentally measured parameters of the memory cells, showed that the transfer to the 55 nm technology, with some improvements of auxiliary CMOS circuits, will allow the implementation of much larger networks with a similar performance lead over the most prospective digital networks [12, 15].

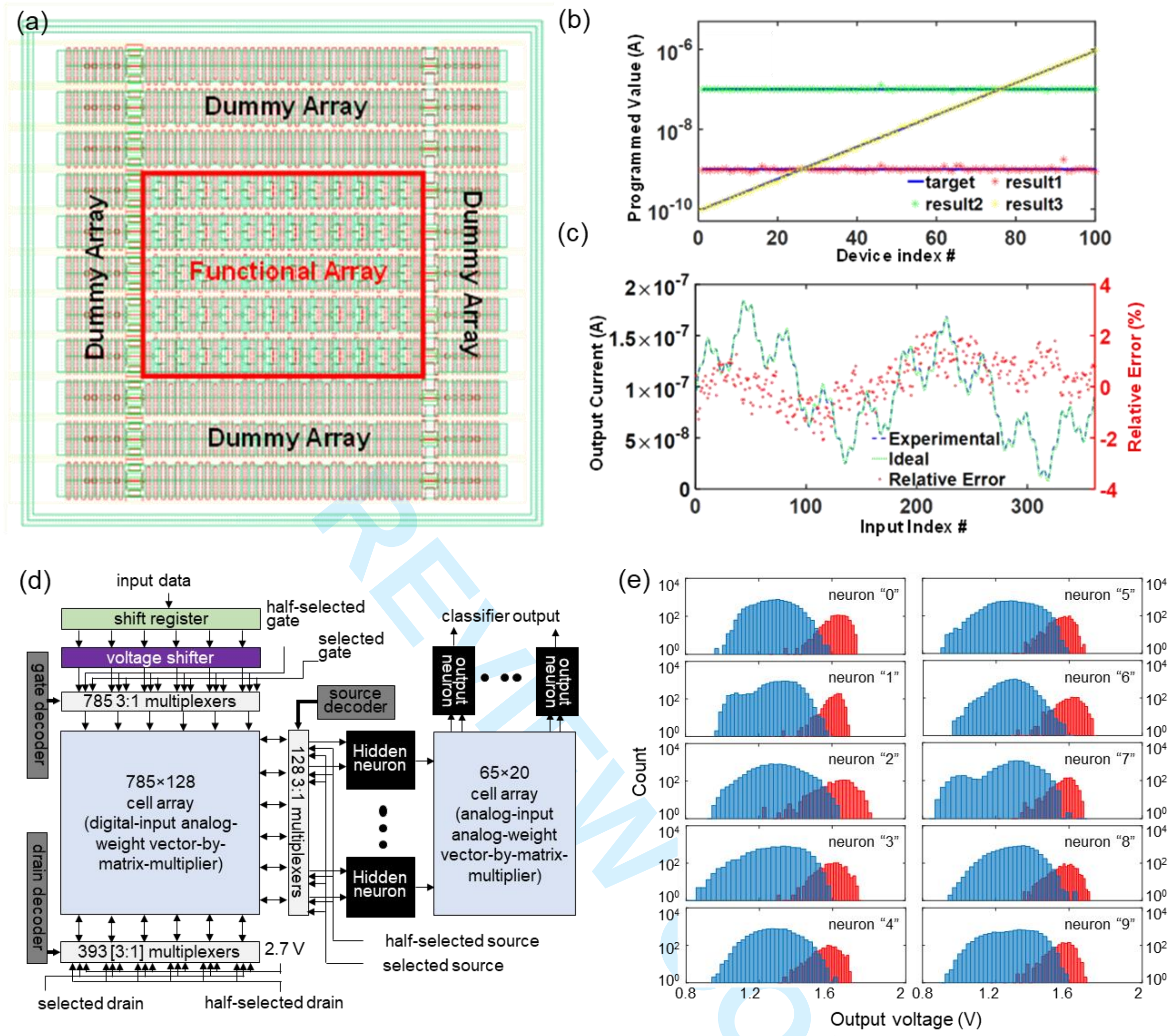


Figure 12 NOR flash memory circuits redesigned for neuro-inspired computing: (a) Layout of a 55-nm vector-matrix multiplication circuit with a 10×(10+2) cell array and auxiliary pass-gates and (b, c) its experimental test results, for (b) cell tuning (measured vs. target weights) and (c) 4-input vector-by-vector multiplication. The four inputs are quasi-DC currents sampled from sine functions with different frequencies. 2-layer MLP based on 180-nm industrial-grade floating-gate devices: (a) high-level architecture (with the weight tuning circuitry for the 2nd array not shown for clarity), and (b) histograms of output voltages for all 10,000 MNIST test patterns. The classification of one pattern takes time below 1 µs time and energy below 20 nJ. Adapted from [107, 108].

5. Device-Circuit-Algorithm Co-Design Perspectives

5.1 Peripheral neuron circuit design considerations

A. Pseudo crossbar array.

The crossbar array is an ideal platform with ultra-high integration density to parallelize the weighted sum and weight update, as discussed in Section 4.1. However, such true crossbar array (without optimized selectors) faces severer cross-talk, IR drop problem and high power consumption as many cells in the true crossbar are half selected (thus conducting current) during the programming. Alternatively, the pseudo-crossbar with 1T1R is widely used as a near-term solution [110], as used in demonstrations in Section 4.2

A and B. Figure 13 (a) shows the pseudo-crossbar array architecture with peripheral supporting circuitry. The key feature of the pseudo-crossbar is that the bit lines (BLs) and source lines (SLs) perpendicular. When all word lines (WLs) are turned on, the transistors in the array are in deep triode region and become transparent, thus SLs and BLs form a pseudo-crossbar. With the help of the transistors, we can select an arbitrary part of the arrays for programming and minimize the IR drop and power consumption of the unselected part. To enable the parallel weighted sum operation, in the pseudo-crossbar, the decoder for the WLs needs a modification from the normal decoder, i.e. adding a NOR gate to have enable signal to bypass and turn on all WLs. In addition, a switch matrix for the BLs (and SLs) is required for simultaneously turn on an arbitrary number of rows (or columns).

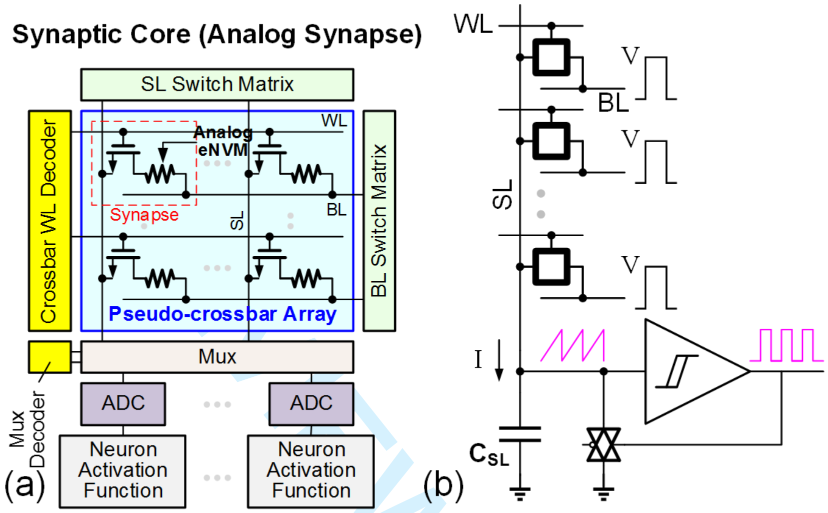


Figure 13 (a) Circuit diagram of the pseudo-crossbar array architecture with peripheral supporting circuitry. When all WLs are turned on, the transistors in the array are in deep triode region and become transparent, thus SLs and BLs form a pseudo-crossbar. (b) The input stage of a neuron node that integrates the analog column current and convert to spikes or digital outputs, serving as an analog-to-digital converter (ADC). Schmitt trigger comparator is typically used.

B. Neuron circuits

As discussed in Section 4.1, when the crossbar array implements the weighted sum, analog current that is proportional to the weighted sum will be sink to the neuron node at the end of each column. The neuron node thus integrates this analog current and convert to spikes or digital outputs before sending to the nonlinear activation function, essentially serving as an analog-to-digital converter (ADC). Depending on the form of nonlinear activation function, different circuit design options could be available. For example, the simple step function could be implemented by a comparator, the rectifying linear function (ReLU) could be implemented by a shift register, and the sigmoid function could be implemented by a look-up-table.

The conventional neuron node design generally employs the integrate-and-fire neuron model for the input stage, as shown in Figure 13 (b): the weighted sum current is integrated in the column capacitance (C_{BL}) and once the membrane voltage (V_{in}) exceeds the threshold voltage (V_p), it triggers a Schmitt trigger comparator circuit to flip and generate the output spike (V_{spike}), while the spike resets the V_{in} by discharging through the V_{reset} path. Figure 14 (a) shows an example of the silicon CMOS neuron design using this principle [111]: Figure 14 (b) shows the simulated waveform of V_{in} and V_{spike} for different weighted sum current (6 μA vs. 1 μA). The number of output spike is designed to be proportional to the amplitude of the input weight sum current. Apparently, such silicon CMOS neuron node is complex and occupies much larger size than the column pitch of the crossbar array. This causes the column pitch matching problem (multiple columns have to share one neuron), thereby reducing the parallelism of the neural networks.

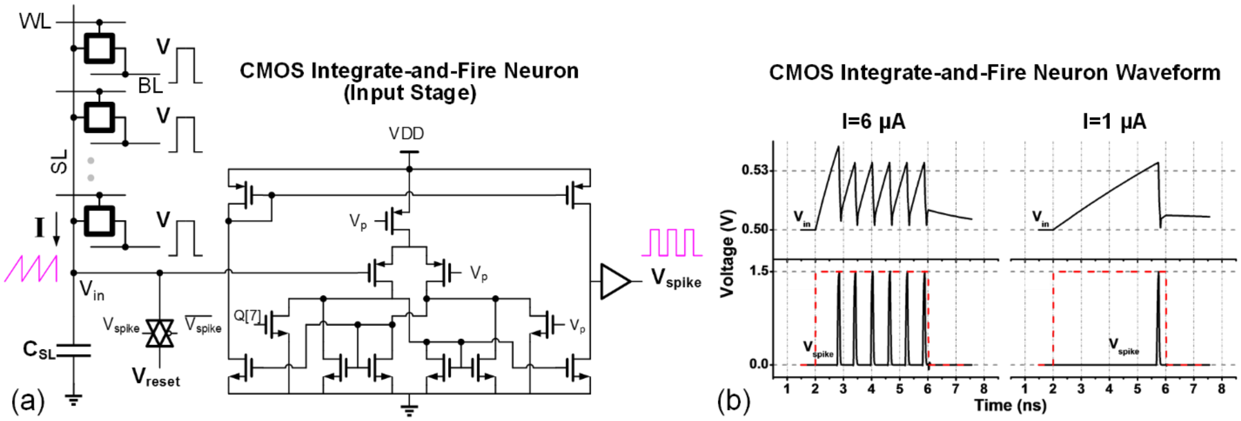


Figure 14 (a) One example of the CMOS integrate-and-fire neuron design for ADC input stage. The membrane voltage integrates and discharges after triggering the output spike. (b) Simulated waveform of the V_{in} and V_{spike} nodes, the output spike frequency is proportional to the input column current. Adapted from [111].

Therefore, it is attractive to design a compact neuron node by using a single device that still function as a Schmitt trigger comparator, as shown in Figure 15 (a). Phase transition in correlated oxides and/or chalcogenides could be exploited due to the volatile threshold switching I-V with hysteresis. The voltage on the phase transition device (if placed at the end of the column) will oscillate and emulate the V_{in} node in the silicon CMOS neuron, behaving as an oscillatory neuron. With a following inverter, the oscillation waveform could be restored to be rail-to-rail (from supply voltage to ground). It is also expected that the oscillation frequency is proportional to the weighted sum current. The circuit-level benchmark work [112] compared the design of silicon CMOS neuron and the oscillatory neuron. These two designs perform exactly the same function: converting the weighted sum current from the column into the number of spikes proportionally. The SPICE simulation shows that the oscillation neuron consumes ~0.033 pJ/spike, while the CMOS neuron consumes 0.168 pJ/spike, leading to a >5X improvement in energy on the neuron node. Oscillatory neuron also shows a >12.5X reduction of the area at the same technology node.

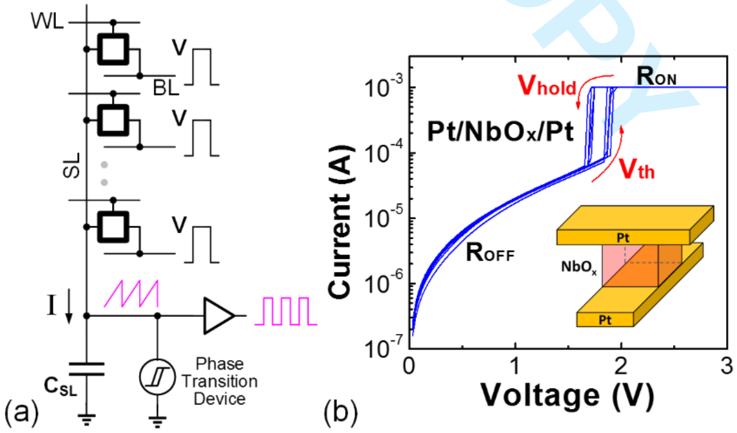


Figure 15 (a) Using a phase transition device at the end of the column to perform the thresholding function, serving as an oscillatory neuron node. Adapted from [112]. (b) The threshold switching I-V characteristics of the NbO_2 device based on meal-insulator-transition mechanism. Adapted from [113].

The NbO_2 based device that exhibits meal-insulator-transition has been proposed as such oscillatory neuron [114]. Figure 15 (b) shows the measured threshold switching I-V characteristics of the Pt/NbO_x/Pt device structure [113]. A hysteresis exists: off-to-on switching threshold voltage (V_{th}) is about 1.9 V and on-to-off switching hold voltage (V_{hold}) is about 1.7 V. To make the neuron node oscillate in a proof-of-concept experiment, the NbO_x device is connected with a load resistor (R_L) as synapse to demonstrate the oscillatory neuron function, as show in Figure 16 (a). The resistance of the load resistor is chosen in between NbO_x device's ON state (R_{ON}) and OFF state (R_{OFF}), and there is a parasitic capacitance at the neuron node. When the read voltage V_R is applied, the membrane voltage on the capacitor will be charged because most of the voltage drop is on the NbO_x device (R_{OFF}>R_L). Once the voltage exceeds V_{th}, the NbO_x device switches to R_{ON}, and the capacitor starts discharging since the voltage drop on the NbO_x device becomes small (R_{ON}<R_L). Once the membrane voltage decreases below V_{hold}, the NbO_x device switches to R_{OFF}. This charging and discharging process repeats, thus the voltage of the neuron node oscillates between V_{hold} and V_{th}. Figure 16 (b), (c), (d) show the measured oscillation frequency with different R_L, i.e. different synaptic weights. A voltage pulse is applied to Channel 1 (CH1) and the node voltage is monitored on Channel 2 (CH2) using the oscilloscope. The oscillation frequency is 2 MHz, 0.7 MHz, and 0.4 MHz with the different load resistance 3.6 KΩ, 11.5 KΩ, and 16.1 KΩ, respectively. This suggests that the oscillation frequency is proportional to the equivalent resistance of the column (i.e. weighted sum results) if connecting the NbO_x neuron to the crossbar array.

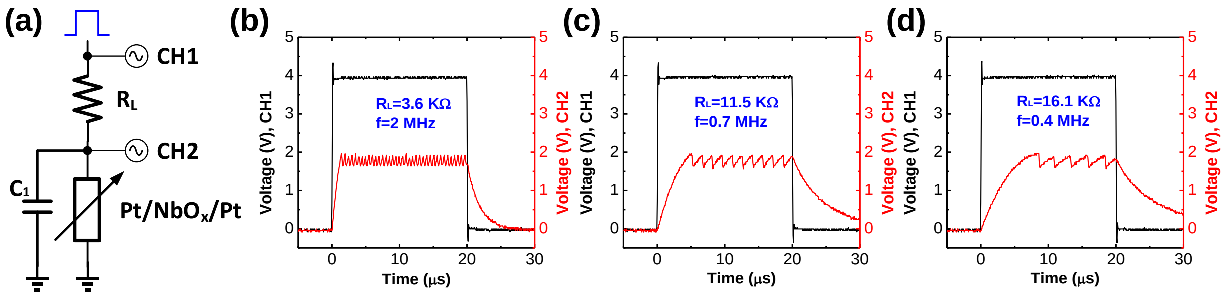


Figure 16 (a) Circuit configuration of an oscillatory neuron node with Pt/NbO_x/Pt device and a load resistor (R_L) as synapse. Oscillation characteristics of with various R_L: (b) R_L=3.6 KΩ, frequency= 2 MHz. (c) R_L=11.5 KΩ, frequency= 0.7 MHz. (b) R_L=16.1 KΩ, frequency= 0.4 MHz. The oscillation frequency is proportional to the synaptic conductance. C1 is estimated to be 573 pF limited by the parasitic capacitance of the electrode pad. Adapted from [113].

5.2 ASU's NeuroSim platform for benchmarking non-ideal device characteristics

Despite of the recent progress in the array-level demonstration of crossbar array with synaptic devices as discussed in Section 4.2, great challenges exist in scaling up the array size to implement large-scale neural networks with high learning accuracy, primarily due to the non-ideal device effects. An ideal weight update behavior of analog synapse assumes a linear update of the conductance (or weight) with programming voltage pulses. As shown in Figure 4, however, representative synaptic devices reported in literature do not follow such ideal trajectory, exhibiting non-ideal properties, including: 1) precision (or number of levels) in the synaptic devices is limited as opposed to the floating-point in software; 2) weight update (conductance vs. # pulse) in today's devices is nonlinear and asymmetric; 3) weight on/off ratio is finite as opposed to the infinity in software, as the off-state conductance is not perfectly zero in realistic devices; 4) device variation, including the *spatial* variation from device to device and the *temporal* variation from cycle to cycle, is remarkable; 5) at array-level, the IR drop along interconnect resistance distorts the weighted sum. These non-ideal behaviors commonly exist in today's synaptic devices and are potentially harmful to the learning accuracy, as indicated by device-algorithm co-simulation studies [49, 51, 90, 115].

Recently, architectural simulator platforms (e.g. PRIME [116], ISAAC [117] and Harmonica [118]) have been developed to support system-level design of neuromorphic accelerators, however they have limited considerations at the aforementioned non-ideal device properties (i.e., they only considered the weight precision and/or variation). On the other hand, MNSIM [119] is a circuit-level macro model of neuro-inspired architecture, but the accuracy in this model is the output error of weighted sum (matrix-vector multiplication), which is just one step of the algorithm thus it lacks the run-time learning accuracy of the entire algorithm. In such context, it is crucial to develop a circuit-level macro model that can be integrated with the learning algorithm (neural network) to form a simulation platform that is hierarchically organized from the device level, the circuit level up to the algorithm level, where each level covers a wide variety of design options.

Here we present a simulation platform "NeuroSim" to evaluate system-level metrics such as learning accuracy, area, latency and energy for online training with these realistic device properties. NeuroSim is a circuit-level macro model implemented in C++ that can be used to estimate the area, latency, dynamic energy and leakage power of on-chip accelerators with SRAM and eNVM arrays. The source code of NeuroSim is available for downloading at [120]. The hierarchy of NeuroSim consists of different levels of abstraction from the memory cell parameters and transistor technology parameters, to the gate-level sub-circuit modules and then to the array architecture including the peripheral circuits. At the device level, important parameters in transistor models include device W/L, the operating and threshold voltage, gate and parasitic capacitance (per unit area) and NMOS/PMOS saturation/off current density, etc. Based on these parameters, the area and intrinsic RC model of standard logic gates (INV, NAND, NOR) can be calculated using analytical equations, thus the circuit-level performance metrics of each sub-circuit module can be estimated.

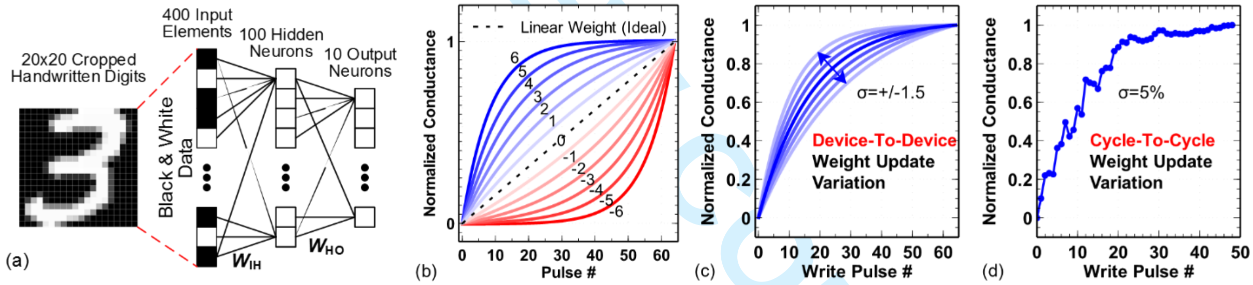


Figure 17 (a) 2-layer MLP network topology with MNIST images as input. (b) Behavior model of weight update in analog synaptic device, with nonlinearity degree labeled from 6 to -6. (c) Device-to-device weight update variation. (d) Cycle-to-cycle weight update variation.

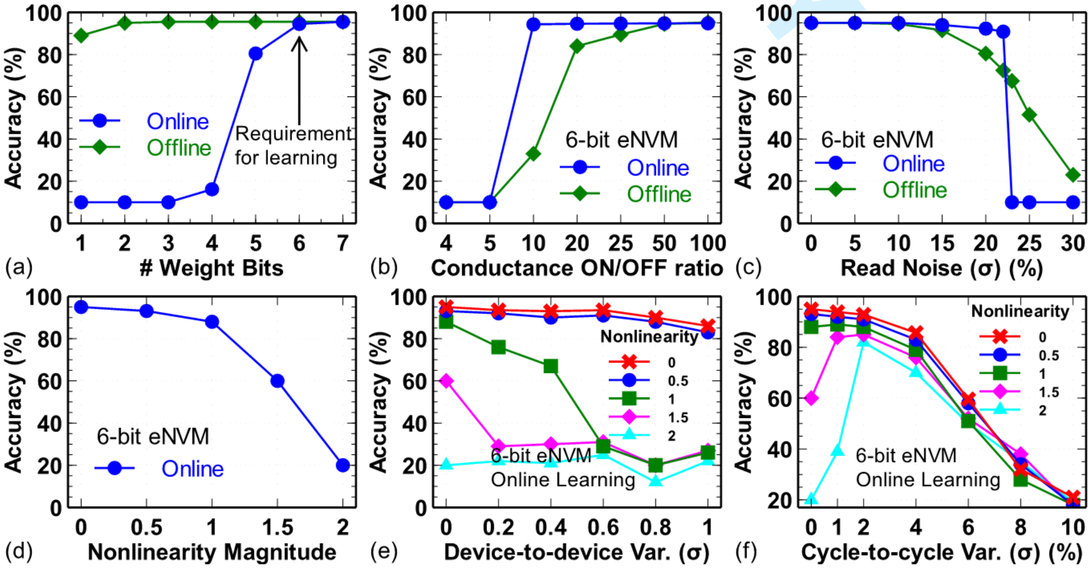


Figure 18 NeuroSim benchmark results of learning accuracy for 2-layer MLP with MNIST dataset. (a) Impact of weight precision, 6-bit is required for online training. (b) Impact of conductance on/off ratio. (c) Impact of read noise of weights. (d) Impact of weight update nonlinearity. Learning accuracy is very sensitive to nonlinearity. (e) Impact of device-to-device variation. (f) Impact of cycle-to-cycle variation.

With NeuroSim circuit-level macro model, an integrated framework could be set-up with any neural network algorithm. As a case study, a 2-layer multilayer perceptron (MLP) with MNIST handwritten dataset is used to benchmark the online training or offline inference capability with synaptic devices. As shown in Figure 17 (a), the MLP network topology is 400(input layer)-100(hidden layer)-10(output layer). Such simple 2-layer MLP can achieve 96~97% in the software baseline. To model the eNVM synaptic properties, a behavior model of analog eNVM cells has been introduced with flexible parameters such as max/min conductance, read/write voltage and pulse width, number of multilevel (precision), and weight update nonlinearity degree (labeled from 6 to -6), as shown in Figure 17 (b). Here the device-to-device variation is defined as the nonlinearity baseline's standard deviation (σ) respect to 1 label of the 6 labels, as shown in Figure 17 (c). For offline training, there is no nonlinearity issue as the cell conductance can be iteratively programmed to the desired value [80, 82]. Cycle-to-cycle variation is referred to as the variation in conductance change at every programming pulse. The cycle-to-cycle variation (σ) is expressed in terms of the percentage of entire weight range, as shown in Figure 17 (d).

Figure 18 shows the sensitivity analysis of each device parameter's effect on learning accuracy. Figure 18 (a) shows that 6-bit weight is required for online learning, while 1 or 2-bit weight may be sufficient for offline inference (at least for MNIST dataset). Figure 18 (b) shows that limited on/off ratio (e.g. <20) will degrade the accuracy because the minimum weight that can be mapped to the device conductance is determined by the on/off ratio. Figure 18 (c) shows that the network has certain tolerance to read noise of the weights up to ~20%. Figure 18 (d) shows that the nonlinearity in the weight update (>1 label) significantly degrade the learning accuracy. Figure 18 (e) shows that the neural network is resilient to the device-to-device variation, except at high nonlinearity in the weight update. Figure 18 (f) shows that small variation (<2%) can alleviate the degradation of learning accuracy by high nonlinearity. The reason may be attributed to the random disturbance that aids convergence of the weights to an optimal weight pattern (i.e. to help the system jump out of local minima in the saturation regime of the nonlinear weight update). However, too large variation (>2%) overwhelms the deterministic update amount defined by the backpropagation thus is harmful to the learning accuracy. At array level, the IR drop problem is also considered. It is estimated that at a wire width of 40 nm, the on-state resistance (R_{ON}, which corresponds to the max conductance state) of eNVM should be higher than 10 kΩ and 500 kΩ to prevent accuracy drop in online learning and inference, respectively. Online learning can also tolerate more IR drop possibly because of the ability for the network to adapt itself to this spatial effect.

Table 3 surveys the representative analog synaptic devices in the literature (as shown in Figure 5 above) with the extracted realistic device parameters such as precision (# of bits), weight update nonlinearity degree, on-state resistance (Ron), on/off ratio, programing pulse condition, and weight update variation, etc. Then the system-level metrics such as learning accuracy, area, latency and energy for online training 1 million MNIST images are listed below. The benchmark results suggest that today's synaptic devices have poor learning accuracy, primarily due to too large weight update nonlinearity (>1), and very limited on/off ratio (<10), etc. In addition, the training latency is too slow, i.e. 10E8 seconds are on the order of years, thus reducing the programming pulse width down to 100 ns or 10 ns is necessary to complete the training within one day or a few hours. Therefore, the targeted and ideal device specifications are listed in the last two columns of Table 3 as guidelines for future device engineering.

Table 3 Extracted device parameters and system-level metrics of the representative analog synaptic devices in the literature (as shown in Figure 4). The simulation is done with NeuroSim platform for online training 1 million MNIST images with a 2-layer MLP neural network.

Device type	TaO_x/TiO_2	PCMO	Ag:a-Si	AlO_x/HfO_2	Target	Ideal
# of bits	6	5	6	5	6	6
Nonlinearity (weight increase/decrease)	1.13/0.72	3.25/5.82	1.13/2.65	3.0/1.0	1.0/1.0	0/0
R_{ON}	5 MΩ	23 MΩ	26 MΩ	16.9 KΩ	200 KΩ	200 KΩ
on/off ratio	2	6.84	12.5	4.43	50	50
Weight increase pulse	3V/40ms	-2V/1ms	3.2V/300 μs	0.9V/100 μs	2V/100ns	2V/10ns
Weight decrease pulse	-3V/10ms	2V/1ms	-2.8V/300 μs	-1V/100 μs	2V/100ns	2V/10ns
Weight update variation (σ)	<1%	<1%	3.50%	5%	2%	0%
Learning accuracy	9.80%	10.09%	76.79%	10.10%	88.66%	94.23%
Area (μm^2)	1.07E+03	1.07E+03	1.07E+03	9.08E+03	1.33E+03	1.33E+03
Latency (s)	2.21E+10	4.34E+08	2.67E+08	4.34E+07	8.82E+04	8.82E+03
Energy (J)	4.36E+04	6.37E+01	6.42E+01	2.09E+03	1.30E+00	1.81E-01

6. Outlook

In the past few years, the neuro-inspired computing with eNVMs has seen remarkable progresses from the single device to array-level demonstrations. Nevertheless, design challenges arise from the device-level up to the architecture-level when the crossbar array size is scaled up to solve practical problems [110]: Firstly, the resistive devices today are mostly engineered towards the digital memory application, but the requirements of synaptic devices are quite different. For instance, synaptic devices need many more multilevel states (up to several hundreds of states) than digital memory's 1 bit to 3 bits (8 levels) thereby requiring special materials and device engineering. Secondly, with the increase of the array size, issues associated with device yield, device variability, and array parasitics show up and may degrade the system performance, while the circuit or architectural mitigation techniques are yet to be developed. In addition to the array core, designs of the peripheral circuits are rarely explored. Furthermore, how to efficiently map various deep learning algorithms into the neuro-inspired architecture is an open question at the architecture-level. EDA tools are necessary for partitioning the array size and the number of layers given the constraints of area, latency, power and learning accuracy. Lastly, the research in this field so far is mostly based on experimental work of single device or small-scale array with software neuron and projection of large-scale array performance by simulations only. A large-scale prototype demonstration with monolithic integration of eNVM devices on top of CMOS and peripheral neuron circuits is critical to make a breakthrough in this field, as one can actually measure the speed and energy efficiency.

Given the challenges mentioned in Section 5.2 and the engineering targets in Table 3, there is still a long way to go for realizing online training on the eNVM devices. Therefore, the most promising application in the near term is the inference-only with offline training. In this case, today's eNVM devices meet most of the requirements as it just needs a good on/off ratio (e.g. 100) to provide sufficient multilevel states (e.g. 100) and a reasonable cycling endurance (e.g. 1000) for iterative programming, although the conductance tuning accuracy (by write-verify) and uniformity across the entire array needs further improvement. The data retention in the intermediate states needs further characterization. To enable the true crossbar array, threshold switching I-V selector that is compatible with the eNVM device properties is critical.

In the next, we will discuss the customization of algorithms point of view to allow efficient hardware implementation. Most of the deep learning algorithms today generally rely on the availability of large datasets and the high-precision training to generate a huge set of model parameters, which are major limitations in mobile and dynamically varying applications. The high precision of data representation needed by deep learning algorithms, which directly impacts the computation cost and energy efficiency. Typically, deep learning models are trained in the GPU environment using 32-bit floating point, in order to satisfy the precision required by backpropagation or other gradient-based approaches; computations with such high-precision data consumes significant amount of hardware resources and is impractical for eNVM devices. Recent research efforts from the algorithm's perspective have made the attempt, including the network pruning [121] and low-precision (fixed-point) training with stochastic rounding of last few bits [122]. The adoption of low-precision weight is most suitable for the feedforward inference stage of the CNN models, including LeNet5 [123], AlexNet [3] and VGG [124] in an order of the complexity. To the extreme case, the binary weight and neuron, namely Binary-Net [125], has been demonstrated for the classification of CIFAR dataset with negligible degradation of accuracy. A similar work that constraints the weights and neurons to be (+1, -1), namely XNOR-Net [126], has been demonstrated for the classification of ImageNet dataset with slight degradation of accuracy. In XNOR-Net, the matrix-vector multiplication essentially becomes the bitwise XNOR operation.

Here, we use a feedforward neural network (applicable to MLP and CNN) as an example to illustrate the hybrid precision requirement for the weight propagation and weight update [127]. Figure 19 shows the flow of the feedforward inference (FF) and backward propagation (BP) for weight update. In the feedforward (FF) inference, the low precision (i.e. 1-bit binary) weights and 1-bit step function neuron could be used for the computation. In the backward propagation (BP), still the low precision weights could be used for calculating the error for weight update. But the weight update should be accumulated on a higher precision, e.g. 6-bit weights (for MNIST dataset). After the 6-bit weights are updated, they could be truncated to 1-bit for the propagation again. We trained binary neural networks on the Theano platform [128]. A MLP with a structure of 784-512-512-512-10 and a CNN with 6 convolution layers and 3 fully-connected layers are trained for evaluations on MNIST and CIFAR-10 datasets, respectively. Table 4 presents the corresponding classification accuracy with floating point (FL) precision and binary precision for these two networks. For MLP on MNIST, the accuracy slightly drops from 99.00% to 98.77%; for CNN on CIFAR-10, the accuracy slightly decreases from 89.98% to 88.47%. This means that as an interim solution before the analog eNVM devices become technologically mature, we could use available binary eNVM devices to prototype large-scale systems to demonstrate practical problems if such accuracy degradation is acceptable for the given application. With this principle, there are a few simulation works to explore the binary neural network with eNVMs [129, 130]. Recently, S. Yu, et al. [41] experimentally demonstrated a binary neural network on a16 Mb RRAM macro chip.

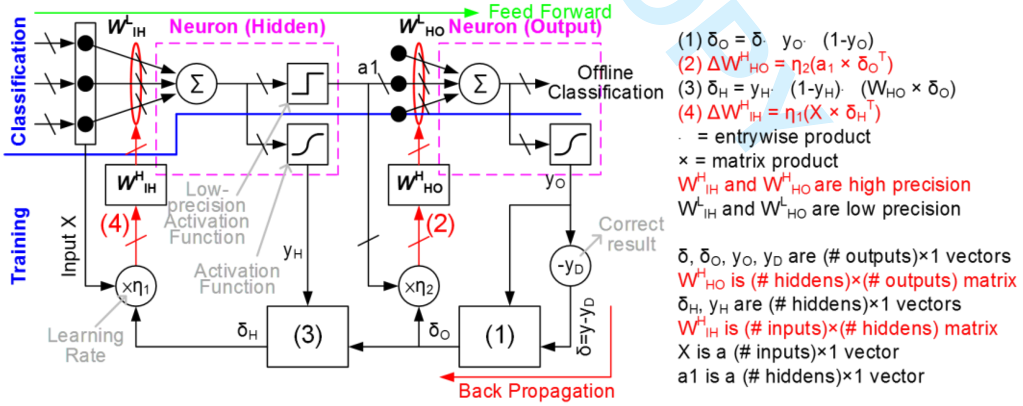
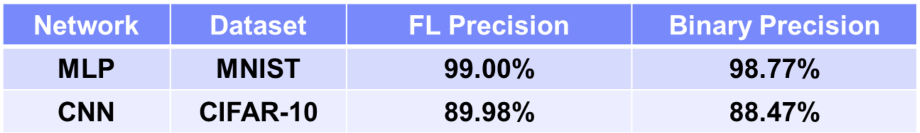

Feed Forward
Classification
W^L_{IH}
Neuron (Hidden)
W^L_{HO}
Neuron (Output)
a1
Offline Classification
Σ
Σ
W^H_{IH}
W^H_{HO}
Low-precision Activation Function
Activation Function
y_O
Correct result
Training
Input X
(4)
(2)
y_H
$\times\eta_1$
$\times\eta_2$
$-y_D$
Learning Rate
δ_H
(3)
δ_O
(1)
$\delta = y_O - y_D$
Back Propagation
(1) $\delta_O = \delta \cdot y_O \cdot (1-y_O)$
(2) $\Delta W^H_{HO} = \eta_2(a_1 \times \delta_O^T)$
(3) $\delta_H = y_H \cdot (1-y_H) \cdot (W_{HO} \times \delta_O)$
(4) $\Delta W^H_{IH} = \eta_1(X \times \delta_H^T)$
· = entrywise product
× = matrix product
W^H_{IH} and W^H_{HO} are high precision
W^L_{IH} and W^L_{HO} are low precision
δ, δ_O, y_O, y_D are (# outputs)×1 vectors
W^H_{HO} is (# hiddens)×(# outputs) matrix
δ_H, y_H are (# hiddens)×1 vectors
W^H_{IH} is (# inputs)×(# hiddens) matrix
X is a (# inputs)×1 vector
a1 is a (# hiddens)×1 vector

Figure 19 The algorithm flow of the feedforward inference (FF) and backward propagation (BP) for weight update in a feedforward neural network. In the FF inference, the low precision (i.e. 1-bit binary) weights and 1-bit step function neuron could be used for the computation. In the backward propagation (BP), still the low precision weights could be used for calculating the error for weight update, but the weight update should be accumulated on a higher precision, e.g. 6-bit weights (for MNIST dataset). Adapted from [127].



Network	Dataset	FL Precision	Binary Precision
MLP	MNIST	99.00%	98.77%
CNN	CIFAR-10	89.98%	88.47%

Table 4 The classification accuracy of MNIST dataset for a MLP and CIFAR-10 dataset for a CNN network. Only slight degradation in accuracy when aggressively truncated to be 1-bit.

Although research recent efforts on network pruning and precision reduction show the promises for the propagation stage, high precision is still a must for the weight update stage due to the gradient descent in the error backpropagation or the decay of learning rate in the machine learning driven algorithms. On the other hand, the biologically-plausible algorithms may naturally tolerate the low precision and variations/noises in the eNVMs evening in the weight update stage. However, the biologically-plausible algorithms have not yet demonstrated a competitively high learning accuracy for solving practical problems. Therefore, the research community should fundamentally re-think the algorithm and hardware co-optimization. Significant cross-layer research efforts are needed to develop new algorithms that can exploit the underlying unique device properties to realize a compact and energy-efficient mapping to the crossbar array architecture.

In the past few years, hardware implementation of neuro-inspired computing has made substantial progresses as summarized in this review. This research also attracted a lot of interests in academic universities and industrial research institutions and companies, as reflected by the large-scale projects such as DARPA SyNAPSE, DARPA UPSIDE, NSF Expeditions in Computing, NSF/SRC E2CDA, SRC/DARPA JUMP in USA, and Human Brain Project (HBP) and NeuRAM3 in Europe, etc. The research on eNVM based synaptic devices, circuits and architectures is highly interdisciplinary in its nature, connecting the fields of materials engineering, nanotechnology, semiconductor device, VLSI design, EDA, computer architecture, machine learning, and computational neuroscience, etc. This review has presented state-of-the-art synaptic device properties, small-scale to medium-scale array integration, and preliminary exploration of device-architecture-algorithm co-design, with the hope of inspiring the research community for the future interdisciplinary collaborations on this emerging and exciting research topic. We anticipate the close interaction between these interdisciplinary fields would lead to a breakthrough in the large-scale demonstration of neuro-inspired computing system in the next decade.

Acknowledgement:

The author wants to thank the technical discussions with Prof. Dmitri Strukov from UCSB, and ASU's student/postdoc researchers who involved in this research project including Pai-Yu Chen, Rui Liu, Xiaoyu Sun, Xiaochen Peng, Zhiwei Li, Ligang Gao, Jiyong Woo and Panni Wang. The author's work is supported in part by NSF-CCF-1552687, NSF-CCF-1715443, NSF-ECCS-1701565, HDTRA1-17-1-0038, NSF/SRC E2CDA, ASCENT (one of the six SRC/DARPA JUMP Centers) and grants from Qualcomm and Samsung.

Author Biography:

Shimeng Yu (S'10-M'14) received the B.S. degree in microelectronics from Peking University, Beijing, China in 2009, and the M.S. degree and Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA in 2011, and in 2013, respectively. He is currently an assistant professor of electrical engineering and computer engineering at Arizona State University, Tempe, AZ, USA. His research interests are emerging nano-devices and circuits with a focus on the resistive memories for different applications including machine/deep learning, neuromorphic computing, hardware security, and radiation-hard electronics, etc. He has published >60 journal papers and >100 conference papers with citations >5000 and

H-index 32. Among his honors, he is a recipient of the Stanford Graduate Fellowship from 2009 to 2012, the IEEE Electron Devices Society Masters Student Fellowship in 2010, the IEEE Electron Devices Society PhD Student Fellowship in 2012, the DOD-DTRA Young Investigator Award in 2015, the NSF Faculty Early CAREER Award in 2016, and the ASU Fulton Outstanding Assistant Professor in 2017. He served the Technical Program Committee for IEEE International Symposium on Circuits and Systems (ISCAS) 2015-2017, ACM/IEEE Design Automation Conference (DAC) 2017-2018, and IEEE International Electron Devices Meeting (IEDM) 2017-2018, etc.

References

[1] D. Silver, et al., "Mastering the game of Go with deep neural networks and tree search," *Nature,* vol. 529, p. 484–489, 2016.

[2] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning," *Nature ,* vol. 521, p. 436–444, 2015.

[3] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012.

[4] A. Graves, A.-r. Mohamed, G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[5] "ImageNet," [Online]. Available: http://www.image-net.org/.

[6] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, A. Ng, "Building high-level features using large scale unsupervised learning," in *International Conference in Machine Learning*, 2012.

[8] S. B. Furber, F. Galluppi, S. Temple, L. A. Plana, "The SpiNNaker project," *Proceedings of the IEEE ,* vol. 102, no. 5, pp. 652-665, 2014.

[9] S. Schmitt, et al., "Neuromorphic hardware in the loop: training a deep spiking network on the BrainScaleS wafer-scale," in *International Joint Conference on Neural Networks (IJCNN)*, 2017.

[10] N. P. Jouppi, et al. , "In-datacenter performance analysis of a tensor processing unit," in *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2017.

[11] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, D. S. Modha , "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science,* vol. 345, no. 6197, pp. 668-673, 2014.

[12] Y-H. Chen, T. Krishna, J. Emer, V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2016.

[13] J. Sim, J-S. Park, M. Kim, D. Bae, Y. Choi, L-S. Kim, "A 1.42TOPS/W Deep Convolutional Neural Network Recognition Processor for Intelligent IoE Systems," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2016.

[14] G. Desoli, N. Chawla, T. Boesch, S.-p. Singh, E. Guidetti, F. D. Ambroggi, T. Majo, P. Zambotti, M. Ayodhyawasi, H. Singh, N. Aggarwal, "A 2.9TOPS/W deep convolutional neural network SoC in FD-SOI 28nm for intelligent embedded systems," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2017 .

[15] B. Moons, R. Uytterhoeven, W. Dehaene, M. Verhelst, "ENVISION: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm FDSOI," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2017.

[16] S. Yu (Ed), Neuro-inspired Computing Using Resistive Synaptic Devices, Springer, 2017.

[17] D. Kuzum, S. Yu, and H.-S. P. Wong, "Synaptic electronics: materials, devices and applications," *Nanotechnology,* vol. 24, p. 382001, 2013.

[18] G. W. Burr, P. Narayanan, R. M. Shelby, S. Sidler, I. Boybat, C. di Nolfo, and Y. Leblebici, "Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: comparative performance analysis (accuracy, speed, and power)," in *IEEE International Electron Devices Meeting (IEDM)*, 2015.

[19] B. Rajendran, Y. Liu, J.-s. Seo, K. Gopalakrishnan, L. Chang, D. J. Friedman, and M. B. Ritter, "Specifications of nanoscale devices and circuits for neuromorphic computational systems," *IEEE Transactions on Electron Devices,* vol. 60, no. 1, pp. 246-253, 2013.

[20] H.-S. P. Wong, S. Raoux, S. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, and K. E. Goodson, "Phase change memory," *Proceedings of the IEEE,* vol. 98, no. 12, p. 2201–2227, 2010.

[21] H.-S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, and M.-J. Tsai, "Metal–oxide RRAM," *Proceedings of the IEEE,* vol. 100, no. 6, p. 1951–1970, 2012.

[22] S. Yu, P.-Y. Chen, "Emerging memory technologies: recent trends and prospects," *IEEE Solid State Circuits Magazine,* vol. 8, no. 2, pp. 43-56, 2016.

[23] H. Wu, X. H. Wang, B. Gao, N. Deng, Z. Lu, B. Haukness, G. Bronner, and H. Qian, "Resistive random access memory for future information processing system," *Proceedings of the IEEE,* 2017.

[24] J J. Yang, D. B. Strukov, D. R. Stewart, "Memristive devices for computing," *Nature Nanotechnology,* vol. 8, no. 1, pp. 13-24, 2013.

[25] G. Indiveri, et al., "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience,* vol. 5, no. 73, pp. 1-23, 2011.

[26] J. Hasler, and B. Marr, "Finding a road map to achieve large neuromorphic hardware systems," *Frontiers in Neuroscience,* vol. 7, no. 118, pp. 1-29, 2013.

[27] S. Furber, "Large-scale neuromorphic computing systems," *Journal of Neural Engineering,* vol. 13, p. 051001, 2016.

[28] D. S. Jeong, I. Kim, M. Zieglerb, and H. Kohlstedtb, "Towards artificial neurons and synapses: a materials point of view," *RSC Advances,* vol. 3, p. 3169, 2013.

[29] D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi, and C. S. Hwang, "Memristors for energy-eficient new computing paradigms," *Advanced Electronic Materials,* vol. 2, p. 1600090, 2016.

[30] A. Sally, "Reflections on the Memory Wall," in *Conference on Computing Frontiers*, 2004.

[31] C.-S. Poon, and K. Zhou, "Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities," *Frontiers in Neuroscience,* vol. 5, no. 108, pp. 1-3, 2011.

[32] S. Chetlur, Cl. Woolley, P. Vandermersch, J. Cohen, and J. Tran, "cuDNN: Efficient Primitives for Deep Learning," arXiv:1410.0759, 2014.

[33] G. Lacey, G. W. Taylor, S. Areibi, "Deep Learning on FPGAs: Past, Present, and Future," http://arxiv.org/abs/1602.04283, 2016.

[34] N. Jouppi, "Google supercharges machine learning tasks with TPU custom chip," 2016. [Online]. Available: https://cloudplatform.googleblog.com/2016/05/Google-supercharges-machine-learning-tasks-with-custom-chip.html.

[35] J. Schemmel, D. Bruderle, A. Grubl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010.

[36] C. Zamarreño-Ramos, L. A. Camuñas-Mesa, J. A. Pérez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, and B, Linares-Barranco, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Frontiers in Neuroscience,* vol. 5, no. 26, pp. 1-22, 2011.

[37] M. Prezioso, F. Merrikh-Bayat, B.D. Hoskins, G.C. Adam, K.K. Likharev, and D.B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature,* vol. 521, pp. 61-64, 2015.

[38] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, W. D. Lu, "Sparse coding with memristor networks," *Nature Nanotechnology,* 2017.

[39] P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S. P. Wong, H. Qian, "Face classification using electronic synapses," *Nature Communications,* vol. 8, p. 15199, 2017.

[40] S. Kim, M. Ishii, S. Lewis, T. Perri, M. BrightSky, W. Kim, R. Jordan, G.W. Burr, N. Sosa, A. Ray, J.-P. Han, C. Miller, K. Hosokawa, and C. Lam, "NVM Neuromorphic Core with 64k-cell (256-by-256) Phase Change Memory Synaptic Array with On-Chip Neuron Circuits for Continuous In-Situ Learning," in *IEEE International Electron Devices Meeting (IEDM)*, 2015.

[41] S. Yu, Z. Li, P.-Y. Chen, H. Wu, B. Gao, D. Wang, W. Wu, H. Qian, "Binary neural network with 16 Mb RRAM macro chip for classification and online training," in *IEEE International Electron Devices Meeting (IEDM)*, 2016.

[42] B. J. Zhu, "Magnetoresistive random access memory: the path to competitiveness and scalability," *Proceedings of the IEEE,* vol. 96, no. 11, p. 1786–1798, 2008.

[43] I. Valov, R. Waser, J. R. Jameson, M. N. Kozicki , "Electrochemical metallization memories—fundamentals, applications, prospects," *Nanotechnology,* vol. 22, p. 254003, 2011.

[44] Y. Choi, I. Song, M-H. Park, H. Chung, S. Chang, B. Cho, J. Kim, Y. Oh, D. Kwon, J. Sunwoo, J. Shin, Y. Rho, C. Lee, M. Kang, J. Lee, Y. Kwon, S. Kim, J. Kim, Y-J. Lee, Q.

Wang, S. Cha, S. Ahn, H. Horii, J. Lee, K. Kim, H. Joo, K. Lee, Y-T. Lee, et al., "A 20nm 1.8V 8Gb PRAM with 40MB/s program bandwidth," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2012.

[45] T.-Y. Liu, T. H. Yan, R. Scheuerlein, Y. Chen, J. K. Lee, G. Balakrishnan, G. Yee, H. Zhang, A. Yap, J. Ouyang, T. Sasaki, S. Addepalli, A. Al-Shamma, C.-Y. Chen, M. Gupta, G. Hilton, S. Joshi, A. Kathuria, V. Lai, D. Masiwal, M. Matsumoto, et al. , "A 130.7mm2 2-layer 32Gb ReRAM memory device in 24nm technology," in *IEEE International Solid-State Circuits Conference*, 2013.

[46] A. Kawahara, R. Azuma, Y. Ikeda, K. Kawai, Y. Katoh, K. Tanabe, T. Nakamura, Y. Sumimoto, N. Yamada, N. Nakai, S. Sakamoto, Y. Hayakawa, K. Tsuji, S. Yoneda, A. Himeno, K. Origasa, K. Shimakawa, T. Takagi, T. Mikawa, and K. Aono, "An 8Mb multi-layered cross-point ReRAM macro with 443MB/s write throughput," in *IEEE International Solid-State Circuits Conference*, 2012.

[47] S. Yu, B. Gao, Z. Fang, H. Y. Yu, J. F. Kang, and H.-S. P. Wong, "Stochastic learning in oxide binary synaptic device for neuromorphic computing," *Frontiers in Neuroscience,* vol. 7, p. 186, 2013.

[48] S. Manan, O. Bichler, D. Querlioz, G. Palma, E. Vianello, D. Vuillaume, C. Gamrat, and B. DeSalvo, "CBRAM devices as binary synapses for low-power stochastic neuromorphic systems: auditory (cochlea) and visual (retina) cognitive processing applications," in *IEEE International Electron Devices Meeting*, 2012.

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

[50] I-T. Wang, C.-C. Chang, L.-W. Chiu, T. Chou, and T.-H. Hou, "3D Ta/TaOx/TiO2/Ti synaptic array and linearity tuning of weight update for hardware neural network applications," *Nanotechnology,* vol. 27, p. 365204, 2016.

[51] G. W. Burr, R. M. Shelby, C. D. Nolfo, J. W. Jang, R. S. Shenoy, P. Narayanan, "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element," in *IEEE International Electron Devices Meeting (IEDM)*, 2014.

[52] "MNIST Handwritten Digits Dataset," [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[53] D. Garbin, O. Bichler, E. Vianello, Q. Rafhay, C. Gamrat, L.Perniola, G. Ghibaudo, B. DeSalvo, "Variability-tolerant Convolutional Neural Network for Pattern Recognition Applications based on OxRAM Synapses," in *IEEE International Electron Devices Meeting (IEDM)*, 2014.

[54] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J. K. Gimzewski, and M. Aono, "Short-term plasticity and long-term potentiation mimicked in single inorganic synapses," *Nature Materials,* vol. 10, no. 8, p. 591–595, 2011.

[55] A. Nayak, T. Ohno, T. Tsuruoka, K. Terabe, T. Hasegawa, J. K. Gimzewski, and M. Aono, "Controlling the synaptic plasticity of a Cu2S gap-type atomic switch," *Advanced Functional Materials,* vol. 22, p. 3606–3613, 2012.

[56] M. Suri, O. Bichler, D. Querlioz, G. Palma, E. Vianello, D. Vuillaume, C. Gamrat, B. DeSalvo, "CBRAM devices as binary synapses for low-power stochastic neuromorphic

systems: auditory (cochlea) and visual (retina) cognitive processing applications," *IEEE International Electron Devices Meeting (IEDM),* 2012.

[57] D. Mahalanabis, H.J. Barnaby, Y. Gonzalez-Velo, M.N. Kozicki, S. Vrudhula, P. Dandamudi, "Incremental resistance programming of programmable metallization," *Solid State Electronics,* vol. 100, pp. 39-44, 2014.

[58] Z. Wang, S. Joshi, S. E. Savel'ev, H. Jiang, R. Midya, P. Lin, Mi. Hu, N. Ge, J. P. Strachan, Z. Li, Q. Wu, M. Barnell, G.-L. Li, H. L. Xin, R. S. Williams, Q. Xia, and J. J. Yang, "Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing," *Nature Materials,* vol. 16, p. 101–108, 2017.

[59] 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, "Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device," *Nanotechnology,* vol. 22, p. 254023, 2011.

[60] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, and H.-S. P. Wong, "An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation," *IEEE Transactions on Electron Devices,* vol. 58, no. 8, pp. 2729-2737, 2011.

[61] S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z.-Q. Wang, Al. Calderoni, N. Ramaswamy, D. Ielmini, "Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM," *IEEE Transactions on Electron Devices,* vol. 63, no. 4, pp. 1508-1515, 2016.

[62] T. Chang, S.-H. Jo, and W. Lu,, "Short-term memory to long-term memory transition in a nanoscale memristor," *ACS Nano,* vol. 5, no. 9, p. 7669–7676, 2011.

[63] C. Du, W. Ma, T. Chang, P. Sheridan, and W. D. Lu, "Biorealistic implementation of synaptic functions with oxide memristors through internal ionic dynamics," *Advanced Functional Materials,* vol. 25, p. 4290–4299, 2015.

[64] S. Kim, C. Du, P. Sheridan, W. Ma, S. H. Choi, and W. D. Lu, "Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity," *Nano Letters,* vol. 15, p. 2203−2211, 2015.

[65] D. Kuzum, R. G. D. Jeyasingh, B. Lee, and H.-S. P. Wong, "Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing," *Nano Letters,* vol. 12, no. 5, p. 2179–2186, 2012.

[66] B. L. Jackson, B. Rajendran, G. S. Corrado, M. Breitwisch, G. W. Burr, R. Cheek, K. Gopalakrishnan, S. Raoux, C. T. Rettner, A. Padilla, A. G. Schrott, R. S. Shenoy, B. Kurdi, C. H. Lam, D. S. Dharmendra, "Nanoscale electronic synapses using phase change devices," *ACM Journal on Emerging Technologies in Computing Systems (JETC),* vol. 9, no. 2, p. 12, 2013.

[67] M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, B. DeSalvo, "Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction," in *IEEE International Electron Devices Meeting (IEDM)*, 2011.

[68] Y. Li, Y. Zhong, L. Xu, J. Zhang, X. Xu, H. Sun, X. Miao, "Ultrafast synaptic events in a chalcogenide memristor," *Scientific Reports,* vol. 3, p. 1619 , 2013.

[69] S. Ambrogio, N. Ciocchini, M. Laudato, V. Milo, A. Pirovano, P. Fantini, and D. Ielmini, "Unsupervised learning by spike timing dependent plasticity in phase change memory (PCM) synapses," *Frontiers in Neuroscience,* vol. 10, no. 56, pp. 1-12, 2016.

[70] O. Bichler, M. Suri, D. Querlioz, D. Vuillaume, B. DeSalvo, and C. Gamrat, "Visual pattern extraction using energy-efficient “2-PCM synapse” neuromorphic architecture," *IEEE Transactions on Electron Devices,* vol. 59, no. 8, pp. 2206-2214, 2012.

[71] S. Yu, B. Gao, Z. Fang, H. Y. Yu, J. F. Kang, and H.-S. P. Wong, "A low energy oxide-based electronic synaptic device for neuromorphic visual system with tolerance to device variation," *Advanced Materials,* vol. 25, no. 12, pp. 1774-1779, 2013.

[72] B. Gao, H. Wu, J. Kang, H. Yu, H. Qian, "Oxide-based analog synapse: physical modeling, experimental characterization, and optimization," in *IEEE International Electron Devices Meeting (IEDM)*, 2016.

[73] W. Wu, H. Wu, B. Gao, N. Deng, S. Yu, H. Qian, "Improving analog switching in HfOx based resistive memory with thermal enhanced layer," *IEEE Electron Device Letters,* vol. 38, no. 8, pp. 1019-1022, 2017.

[74] J. Woo, K. Moon, J. Song, S. Lee, M. Kwak, J. Park, and H. Hwang, "Improved synaptic behavior under identical pulses using AlOx/HfO2 bilayer RRAM array for neuromorphic systems," *IEEE Electron Device Letters,* vol. 37, no. 8, pp. 994-997, 2016.

[75] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Letters,* vol. 10, no. 4, p. 1297–1301, 2010.

[76] S. Park, H. Kim, M. Choo, J. Noh, A. Sheri, S. Jung, K. Seon, J. Park, S. Kim, W. Lee, J. Shin, D. Lee, G. Choi, J. Woo, E. Cha, C. Park, M. Jeon, B. Lee, B. H. Lee, and H. Hwang, "RRAM-based synapse for neuromorphic system with pattern recognition function," in *IEEE International Electron Devices Meeting (IEDM)*, 2012.

[77] S. Park, A. Sheri, J. Kim, J. Noh, J. Jang, M. Jeon, B. Lee, B. R. Lee, B. H. Lee, and H. Hwang, "Neuromorphic speech systems using advanced ReRAM-based synapse," in *IEEE International Electron Devices Meeting*, 2013.

[78] L. Gao, I-T. Wang, P.-Y. Chen, S. Vrudhula, J.-S. Seo, Y. Cao, T.-H. Hou, S. Yu, "Fully parallel write/read in resistive synaptic array for accelerating on-chip learning," *Nanotechnology,* 2015.

[79] I-T. Wang, Y.-C. Lin, Y.-F. Wang, C.-W. Hsu, and T.-H. Hou, "3D synaptic architecture with ultralow sub-10 fJ energy per spike for neuromorphic computation," in *IEEE International Electron Devices Meeting*, 2014.

[80] F. Alibart, L. Gao, B. D Hoskins, D. B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," *Nanotechnology,* vol. 23, p. 075201, 2012.

[81] L. Gao, F. Alibart, and D.B. Strukov, "Analog-input analog-weight dot-product operation with Ag/a-Si/Pt memristive devices," in *IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2012.

[82] L. Gao, and S. Yu, "Programming protocol optimization for analog weight tuning in resistive memories," in *IEEE Device Research Conference (DRC)*, 2015.

[83] H. Mulaosmanovic, J. Ocker, S. Müller, M. Noack, J. Müller, P. Polakowski, T. Mikolajick, S. Slesazeck, "Novel ferroelectric FET based synapse for neuromorphic systems," in *IEEE Symposium on VLSI Technology*, 2017.

[84] S. Oh, T. Kim, M. Kwak, J. Song, J. Woo, S. Jeon, I. K. Yoo, H. Hwang, "HfZrOx-based ferroelectric synapse device with 32 levels of conductance states for neuromorphic applications," *IEEE Electron Device Letters,* vol. 38, no. 6, pp. 732-735, 2017.

[85] M. Jerry, P.-Y. Chen, J. Zhang, P. Sharma, K. Ni, S. Yu, S. Datta, "Ferroelectric FET analog synapse for acceleration of deep neural network training," in *IEEE International Electron Devices Meeting (IEDM)*, 2017.

[86] M. Hu, H. Li, Q. Wu, G. S. Rose, "Hardware realization of BSB recall function using memristor crossbar arrays," in *Design Automation Conference*, 2012.

[87] P.-Y. Chen, D. Kadetotad, Z. Xu, A. Mohanty, B. Lin, J. Ye, S. Vrudhula, J.-S. Seo, Y. Cao, S. Yu, "Technology-design co-optimization of resistive cross-point array for accelerating learning algorithms on chip," in *IEEE Design, Automation & Test in Europe (DATE)*, 2015.

[88] B. Liu, H. Li, Y. Chen, X. Li, T. Huang, Q. Wu, and M. Barnell, "Reduction and IR-drop compensations techniques for reliable neuromorphic computing systems," in *IEEE/ACM International Conference on Computer-Aided Design*, 2014.

[89] S. H. Jo, T. Kumar, S. Narayanan, W. D. Lu, and H. Nazarian, "3D-stackable Crossbar Resistive Memory based on Field Assisted Superlinear Threshold (FAST) Selector," in *IEEE International Electron Devices Meeting (IEDM)*, 2014.

[90] T. Gokmen, and Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: design considerations," *Frontiers in Neuroscience,* vol. 10, no. 333, pp. 1-13, 2016.

[91] S. Yu, B. Gao, Z. Fang, H. Y. Yu, J. F. Kang, H.-S. P. Wong, "A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: experimental characterization and large-scale modeling," in *IEEE International Electron Devices Meeting (IEDM)*, 2012.

[92] K.-H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, and W. Lu, "A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications," *Nano Letters,* vol. 12, no. 1, p. 389–395, 2011.

[93] S. B Eryilmaz, D. Kuzum, R. G. D. Jeyasingh, S. Kim, M. BrightSky, C. Lam, and H.-S. P. Wong, "Experimental demonstration of array-level learning with phase change synaptic devices," in *IEEE International Electron Devices Meeting*, 2013.

[94] S. B. Eryilmaz, E. Neftci, S. Joshi, S. Kim, M. BrightSky, H.-L. Lung, C. Lam, G. Cauwenberghs, H.-S. P. Wong, "Training a probabilistic graphical model with resistive switching electronic synapses," *IEEE Transactions on Electron Devices,* vol. 63, no. 12, pp. 5004-5011, 2016.

[95] S. Park, M. Chu, J. Kim, J. Noh, M. Jeon, B. H. Lee, H. Hwang, B. Lee, and B.-G. Lee, "Electronic system with memristive synapses for pattern recognition," *Scientific Reports,* vol. 5, p. 10123, 2015.

[96] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, "Dot-product engine for neuromorphic computing: programming 1T1M

crossbar to accelerate matrix-vector multiplication," in *ACM/IEEE Design Automation Conference (DAC)*, 2016.

[97] L. Gao, P.-Y. Chen, S. Yu, "Demonstration of convolution kernel operation on resistive cross-point array," *IEEE Electron Device Letters,* vol. 37, no. 7, pp. 870-873, 2016.

[98] J. Lu, S. Young, I. Arel, J. Holleman, "A 1 TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13 µm CMOS," *IEEE Journal of Solid-State Circuits,* vol. 50, no. 1, pp. 270-281, 2015.

[99] G. W. Burr, R. M. Shelby, S. Sidler, C. di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. Kurdi and H. Hwang, "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses), using phase change memory as the synaptic weight element," *IEEE Transactions on Electron Devices,* vol. 62, no. 11, pp. 3498-3507, November 2015.

[100] "Yale Face Database," [Online]. Available: http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html.

[101] M. Prezioso, I. Kataeva, F. Merrikh-Bayat, B. Hoskins, G. Adam, T. Sota, K. Likharev, and D. Strukov, "Modeling and implementation of firing-rate neuromorphic-network classifiers with bilayer Pt/Al2O3/TiO2-x/Pt memristors," in *IEEE International Electron Devices Meeting (IEDM)*, 2015.

[102] B. Govoreanu, G. S. Kar, Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. P. Radu, L. Goux, S. Clima, R. Degraeve, N. Jossart, O. Richard, T. Vandeweyer, K. Seo, P. Hendrickx, G. Pourtois, H. Bender, L. Altimime, D. J. Wouters, J. A. Kittl and M. Jurczak, "10×10nm2 Hf /HfOx crossbar resistive RAM with excellent performance , reliability and low-energy operation," in *IEEE International Electron Devices Meeting (IEDM)*, 2011.

[103] G. C. Adam, B. D. Hoskins, M. Prezioso, F. Merrikh-Bayat, B. Chakrabarti, and D.B. Strukov, "3-D memristor crossbars for analog and neuromorphic computing applications," *IEEE Transactions on Electron Devices,* vol. 64, no. 1, pp. 312-318, 2017.

[104] B. Chakrabarti, M. A. Lastras-Montano, G. Adam, M. Prezioso, B. Hoskins, K.-T. Cheng, and D. B. Strukov, "A multiply-add engine with monolithically integrated 3D memristor crossbar/CMOS hybrid circuit," *Scientific Reports,* vol. 7, p. 42429, 2017.

[105] F. Merrikh Bayat, X. Guo, H.A. Om'mani, N. Do, K.K. Likharev, and D.B. Strukov, "Redesigning commercial floating-gate memory for analog computing applications," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015.

[106] F. Merrikh Bayat, X. Guo, M. Klachko, N. Do, K. Likharev, and D. Strukov, "Model-based high-precision tuning of NOR flash memory cells for analog computing applications," in *IEEE Device Research Conference (DRC)*, 2016.

[107] X. Guo, F. Merrikh Bayat, M. Prezioso, Y. Chen, B. Nguyen, N. Do, and D. B. Strukov, "Temperature-insensitive analog vector-by-matrix multiplier based on 55 nm NOR flash memory cells," in *IEEE Custom Integrated Circuits Conference (CICC)*, 2017.

[108] X. Guo, F. Merrikh Bayat, M. Bavandpour, M. Klachko, M. R. Mahmoodi, M. Prezioso, K. K. Likharev, and D. B. Strukov, "Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded NOR flash memory technology," in *IEEE International Electron Devices Meeting (IEDM)*, 2017.

[109] S. K. Esser, R. Appuswamy, P. Merolla, J. V. Arthur, and D. S. Modha, "Backpropagation for energy-efficient neuromorphic computing," in *Conference on Neural Information Processing Systems (NIPS)*, 2015.

[110] S. Yu, P.-Y. Chen, Y. Cao, L. Xia, Y. Wang, H. Wu, "Scaling-up Resistive Synaptic Arrays for Neuro-inspired Architecture: Challenges and Prospect," in *IEEE International Electron Devices Meeting*, 2015.

[111] D. Kadetotad, Z. Xu, A. Mohanty, P.-Y. Chen, B. Lin, J. Ye, S. Vrudhula, S. Yu, Y. Cao, and J.-S. Seo, "Parallel architecture with resistive crosspoint array for dictionary learning acceleration," *IEEE Jounral of Emerging Selected Topics in Circuits and Systems,* vol. 5, no. 2, pp. 194-204, 2015.

[112] P.-Y. Chen, J.-S. Seo, Y. Cao, S. Yu, "Compact oscillation neuron exploiting metal-insulator-transition for neuromorphic computing," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016.

[113] L. Gao, P.-Y. Chen, S. Yu, "NbOx based oscillation neuron for neuromorphic computing," *Applied Physics Letters,* vol. 111, p. 103503, 2017.

[114] K. Moon, E. Cha, J. Park, S. Gi, M. Chu, K. Baek, B. Lee, S. Oh, H. Hwang, "High density neuromorphic system with Mo/Pr0. 7Ca0. 3MnO3 synapse and NbO2 IMT oscillator neuron," in *IEEE International Electron Devices Meeting (IEDM)*, 2015.

[115] S. Agarwal, S. J. Plimpton, D. R. Hughart, A. H. Hsia, I. Richter, J. A. Cox, C. D. James, M. J. Marinella, "Resistive memory device requirements for a neural algorithm accelerator," in *International Joint Conference on Neural Networks (IJCNN)*, 2016.

[116] P. Chi, S. Li, Z. Qi, P. Gu, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "PRIME: A novel processing-In-memory architecture for neural network computation in ReRAM-based main memory," in *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2016.

[117] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, V. Srikumar, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2016.

[118] X. Liu, M. Mao, B. Liu, B. Li, Y. Wang, H. Jiang, M. Barnell, Q. Wu, J. Yang, H. Li, and Y. Chen, "X. Liu, M. Mao, B. Liu, B. Li, Y. Wang, H. Jiang, M. Barnell, Q. Wu, J. Yang, H. Li, and Y. Chen, “Harmonica: A Framework of Heterogeneous Computing systems with memristor-based neuromorphic computing accelerators," *IEEE Transactions on Circuits and Systems I,* vol. 63, no. 5, pp. 617-628, 2016.

[119] L. Xia, B. Li, T. Tang, P. Gu, X. Yin, W. Huangfu, P.-Y. Chen, S. Yu, Y. Cao, Y. Wang, Y. Xie, H. Yang, "MNSIM: Simulation platform for memristor-based neuromorphic computing system," in *IEEE/ACM Design Automation and Test in Europe (DATE)*, 2016.

[120] "NeuroSim," [Online]. Available: https://github.com/neurosim/MLP_NeuroSim.

[121] S. Han, H. Mao, W. J. Dally, "A deep neural network compression pipeline: Pruning, quantization, huffman encoding," in *International Conference on Learning Representations (ICLR)*, 2016.

[122] S. Gupta, A. Agrawal, K. Gopalakrishnan, P. Narayanan, "Deep Learning with Limited Numerical Precision," in *International Conference on Machine Learning (ICML)*, 2015.

[123] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, November 1998.

[124] K. Simonyan and K. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[125] M. Courbariaux, Y. Bengio, J. P. David, "BinaryConnect: Training Deep Neural Networks with binary weights during propagations," in *Advances in Neural Information Processing Systems*, 2015.

[126] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision (ECCV)*, 2016.

[127] Z. Li, P.-Y. Chen, H. Xu, S. Yu, "Design of ternary neural network with 3D vertical RRAM array," *IEEE Transactions on Electron Devices,* vol. 64, no. 6, pp. 2721-2727, 2017.

[128] "Theano," [Online]. Available: https://github.com/Theano/Theano.

[129] T. Tang, L Xia, B. Li, Y. Wang, H. Yang, "Binary convolutional neural network on RRAM," in *IEEE/ACM Asia and South PacificDesign Automation Conference (ASP-DAC)*, 2017.

[130] L. Ni, Z. Liu, W. Song, J. J. Yang, H. Yu, K. Wang, Y. Wang, "An energy-efficient and high-throughput bitwise CNN on sneak-path-free digital ReRAM crossbar," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2017.