Chapter 19

Neuromorphic computing with resistive switching memory devices

Daniele Ielmini[1] and Stefano Ambrogio[2]

[1]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano and IU.NET, Piazza L. da Vinci 32 – 20133 Milano, Italy. Email: daniele.ielmini@polimi.it

[2]IBM Research-Almaden, 650 Harry Road, 95120 San Jose, CA, USA: stefano.ambrogio@ibm.com

Abstract: Brain-inspired computing has been the subject of an intense research in the last decades, with the aim of recreating some of the cognitive computing functions of the human brain in silicon-based hardware. In this frame, resistive switching memories (RRAM) and other emerging memory technologies are extremely promising as they offer memory and plasticity with high scaling capability, thus enabling the integration of a high density of synapses and neurons. This chapter summarizes the status and challenges of RRAM-based neuromorphic engineering. RRAM synapses are described within the frame of various neural network architectures, such as artificial neural networks (ANNs) and spiking neural networks (SNNs). First, ANNs with deep neural network architectures are described in terms of their operation during inference and learning, referring to the typical backpropagation scheme for supervised training. The challenges for high-density, high-functionality ANNs for computer vision with RRAM synapses are addressed. RRAM circuits enabling spike-timing dependent plasticity (STDP) and their use for unsupervised learning in feed-forward and recurrent networks are then presented. A hardware SNN for unsupervised learning by STDP in RRAM synapses is illustrated, demonstrating learning of static and dynamic patterns with both binary and gray-scale values. Finally, an outlook on the prospects of RRAM for future cognitive computing is given.

Keywords: Neuromorphic engineering, resistive switching memory (RRAM), memristor, multilevel storage, deep learning, spike-timing dependent plasticity (STDP), artificial neural network (ANN), spiking neural network (SNN), pattern recognition.

1. Introduction

For the last 50 years, digital computing machines have improved their performance by scaling down the field-effect transistor (FET) according to the Moore's law, which predicts the doubling of the transistor count on the chip every 18 months [1]. Currently, the scaling trend in the microelectronics industry is facing the hard challenge of extreme dynamic power consumption, which is preventing the frequency increase due to the excessive chip heating [2]. To overcome the current limitations of scaling, novel devices have been proposed with the purpose of improving the subthreshold slope, thus allowing for the reduction of the dynamic power at constant static power consumption. New proposals include the tunnel FET [3], the negative-capacitance FET [4], and many other concepts [5], which are currently at the stage of demonstration in academic and industrial research labs.

On the other hand, the search for increasingly high operating frequency is questioned by the energy efficiency of the von Neumann architecture. Fig. 1 shows the power density of commercial central processing unit (CPU) chips as a function of the frequency bandwidth, demonstrating the relentless increase according to the Moore's law since 1971 [6]. In comparison, the human brain is located at the end of low power (about 20 W total consumed power) and low frequency (about 10 Hz). These data indicate that functionality may be improved not only by a mere increase of the number of operations per second, but also, and most importantly, by the architecture of the computing and memory elements.

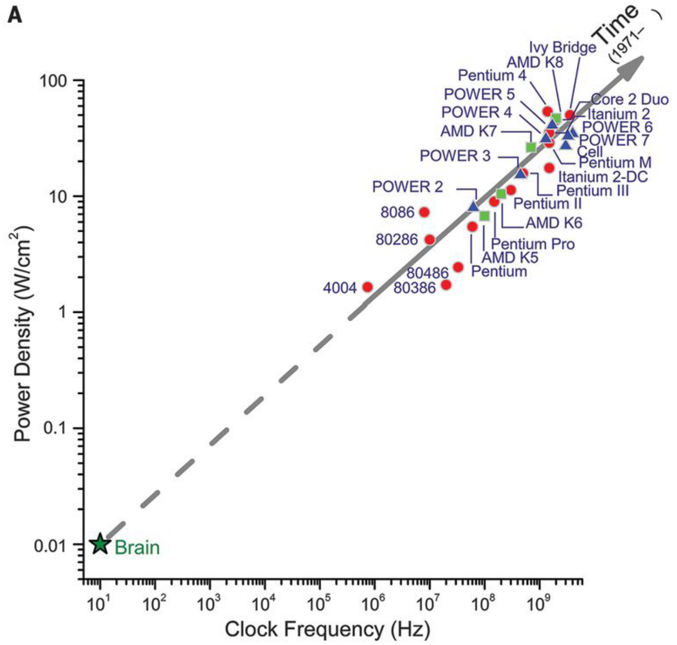

Fig. 1. Power density and clock frequency of digital microprocessors between 1971 and 2014. Moore's law improves both power density and clock frequency, however battery- and thermal-limitations of power dissipation forced a saturation during the last few technology generations. Despite the low frequency and low power density, the brain can outperform a digital computer in many cognitive tasks. Reproduced with permission from [6]. Copyright AAAS (2014).

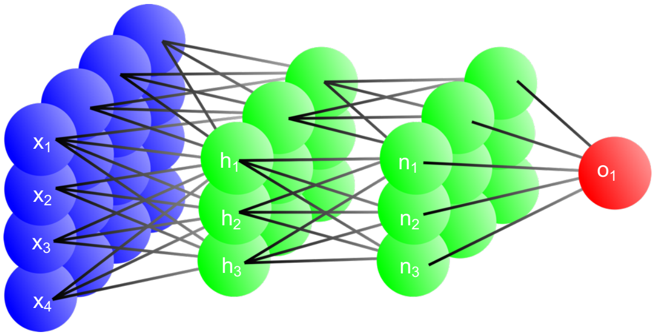

Fig. 2. Sketch of a feed-forward FC neural network with an input layer of neurons (x_1, x_2, etc.), 2 hidden layers (h_1, h_2, …, n_1, n_2, …), and an output layer with one neuron (o_1) for classification.

The high performance and energy efficiency in the human brain can be explained by the co-location of the computing and memory elements, and the parallelism of the neural network architecture. Fig. 2 shows a simple feed-forward neural network with an input layer of neurons, 2 hidden layers, and an output layer with one neuron for classification. The network is fully connected (FC), in that each neuron in layer *i* is connected to each neuron in layer *i-1* and layer *i+1* through synapses. Each neuron in layer *i* collects the signals delivered by neurons in layer *i-1*, weighted by the corresponding synapse. The specific function of the neural network, *e.g.*, the type of pattern that can be recognized and distinguished by the network, is dictated by the weight of the synapses and the architecture, *e.g.*, the number of layers and the number of neurons in each layer. Synapses thus play a critical role for neuromorphic engineering, that is the study and design of neural networks for developing cognitive circuits emulating the functioning of the brain. Note that memory and computing are sparsely distributed within a neural network, thus enabling enhanced performance with respect to von Neumann architectures, where the memory chip and the CPU are physically separated [7].

In this scenario, emerging memory technologies such as resistive switching memory (RRAM) [8-10], phase change memory (PCM) [11-13], and magnetic memory (MRAM) [14-16] can play a pivotal role, thanks to possibility of multilevel storage, nanoscale dimension, back-end-of-line (BEOL) integration, and good reliability. Emerging memory devices can both store data and provide computing functionality, such as Boolean logic operations in RRAM devices [17-20] and PCM devices [21-23]. Accumulative crystallization in PCM allows for algebraic summation [24] and prime factorization [25]. Finally, RRAM and PCM allow for integration in crosspoint arrays, which naturally provide matrix-vector multiplication (MVM) via physical computing through the Ohm's law and the Kirchhoff's law [26, 27].

As a result of the high computing capabilities in emerging memories, several schemes were proposed for artificial synapses and neurons in neuromorphic circuits. Synapses in crosspoint arrays were demonstrated using PCM [28, 29] and RRAM [30, 31]. Brain-inspired synapses capable of changing their weights according to spike-timing dependent plasticity (STDP) were demonstrated by PCM devices [32-34] and RRAM devices [35-40]. Electronic neuron circuits with integrate-and-fire operation were shown either relying on the accumulation of input spiking signals by crystallization in PCM devices [41] or using the threshold switching operation in volatile-type RRAM [42, 43]. Hardware implementations of neurons and synapses with PCM or RRAM technologies were demonstrated for supervised learning [29-31] or unsupervised learning [44]. Despite several challenges to upscale the proposed concepts to the higher level of complete systems capable of cognitive computing, the reported results are extremely promising for the development of neuromorphic circuits based on emerging memories.

This chapter addresses RRAM devices, and most generally emerging memories such as PCMs, for neuromorphic circuits. RRAM-based neural networks will be reviewed with reference to supervised learning and unsupervised learning, discussing RRAM synaptic structures and the techniques for weight update in the network. Brain-inspired associative memory and error correction in recurrent Hopfield networks will be finally discussed.

2. Neural networks for supervised learning

Neural networks find extensive applications in pattern recognition, such as recognition of objects or faces within a picture. The most popular approach to this purpose is the deep learning concept, where FC-ANNs with multiple layer perceptron (MLP) structure are trained to provide an abstract representation of the submitted data [45]. Deep learning has recently led to several breakthroughs in various fields, including speech recognition [46], image recognition and object detection [47], machine translation [48], drug discovery and genomics [49]. The use of 2-terminal memory elements, such as PCM or RRAM, for the synaptic connections in the ANN of Fig. 2 has been early recognized as an attractive strategy for at least 2 reasons: first, the memory element can represent a multiple-bit value thanks to the analog nature of PCM and RRAM, thus offering the capability to replace many single-bit RAM cells for weight storage. For instance, up to 8 levels, *i.e.*, 3 bits, have been demonstrated with RRAM devices [50, 51] and PCM devices [52], thus supporting the feasibility of a very high synaptic density in the ANN. The second added value of using 2-terminal resistive memories is the crosspoint-architecture which enables physical computation of MVM by Ohm’s and Kirchhoff’s laws [26, 27]. In fact, the total current I_j of the crosspoint column of index j is given by:

$$I_j = \sum_i G_{ij} V_i \tag{1}$$

where V_i is the voltage applied to the crosspoint row of index i, and G_{ij} is the synaptic conductance (or weight) connecting the row i and the column j. As a result, MVM is completed in-situ within one single step, without any need for multiplication-accumulation (MAC) requiring multiple steps in a digital CPU with time-consuming exchange of input/output data with the memory chip. Supervised training supported by RRAM/PCM synaptic arrays thus seems a promising trend for saving power and speeding up the learning process for deep learning applications.

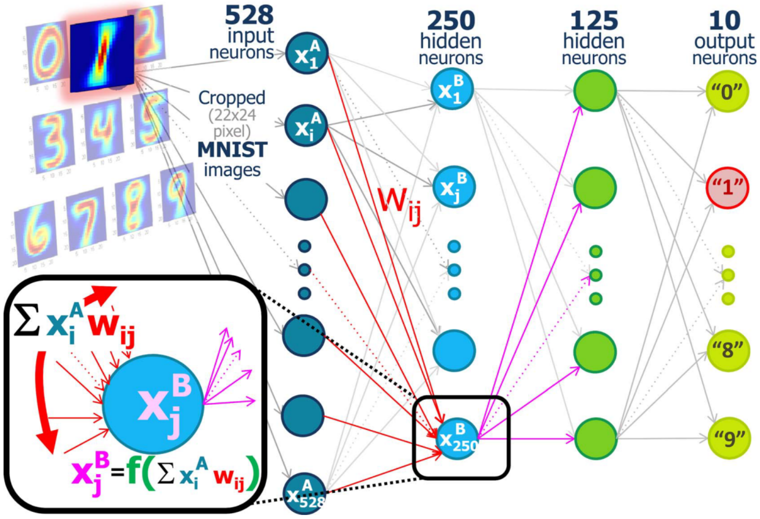

Fig. 3. Sketch of a feedforward FC ANN with MLP structure adopting PCM synapses. For supervised learning of handwritten digits from the MNIST dataset, patterns are submitted to input neurons, then propagated forward by the non-linear transfer function of the neuron according to Eq. (2). Only the output neuron “1” should fire in response to

the presentation of a digit “1” as input pattern. Reproduced with permission from [29]. Copyright IEEE (2015).

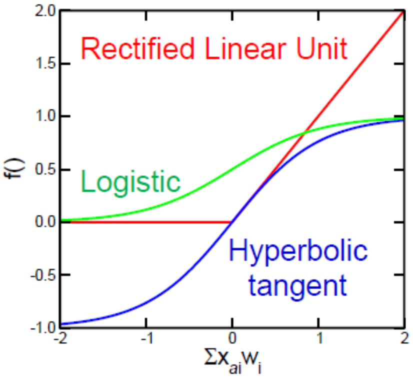

Fig. 4. Examples of non-linear transfer functions adopted for neurons in ANNs, including the hyperbolic tangent function, the logistic function, and the rectifying linear unit (ReLU) function.

2.1 Network training by the backpropagation algorithm

Fig. 3 shows a typical ANN with MLP structure adopting PCM synapses [29]. For supervised learning, first a pattern, *e.g.*, a handwritten digit from the MNIST dataset, is submitted to the input layer, thus resulting in a feedforward propagation of the pattern across the several layers of neurons. In the feedforward propagation, neurons compute their output values as a proper function of the sum of the signals from the preceding layer, each signal multiplied by the corresponding synaptic weight, namely:

$$x_j^B = f\left(\sum_i w_{ij} x_i^A\right) \tag{2}$$

where x_i^A and x_j^B are the output signals of neuron i in layer A and neuron j in layer B, respectively, w_{ij} is the weight of the synapse connecting these 2 neurons, and f is a suitable non-linear function representing the threshold-type behavior of the McCulloch-Pitts (MCP) neuron [53]. Fig. 4 shows some typical transfer functions adopted for ANNs, including the hyperbolic tangent function, the logistic function, and the rectifying linear unit (ReLU) function, which has been demonstrated to speed up the supervised learning in the backpropagation scheme and simplify the tuning of the parameters [54]. At the end of the forward propagation process, the synaptic weights are updated according to the backpropagation scheme shown in Fig. 5. Here, the output signals x_j^D are compared with the correct answer g_j provided by each pattern label, and the comparison leads to an error term δ_j^D given by:

$$\delta_j^D = f'(x_j^D)(g_j - x_j^D) \tag{3}$$

where f' is the derivative of the neuron transfer function with respect to the neuron value x_j^D. The error term δ_j^D in Eq. (3) is then used to update the weights of synapses connected to the output neuron layer according to the incremental update formula [55]:

$$\Delta w_{ij} = \eta x_i^C \delta_j^D \tag{4}$$

where η is a learning efficiency parameter dictating the speed of update of the backpropagation process. Note that the weight in Eq. (4) is changed proportionally to the error δ_j^D, which marks the distance of the actual weight from the ideal value for correctly recognizing the submitted pattern, and the input signal x_i^C, which is a figure of the importance of the synapse in the recognition process for the specific pattern. The learning efficiency η plays a key role for convergence and accuracy, therefore its value must be carefully tuned to maximize the network ability to classify unknown input images. Errors in the preceding hidden layer C are calculated according to [55]:

$$\delta_k^C = f'(x_k^C) \sum_j w_{ij} \delta_i^D \tag{5}$$

where f' is the derivative of the neuron transfer function with respect to the internal variable x_k. Also, Eq. (5) allows to iteratively compute errors across the ANN layers in the backpropagation direction.

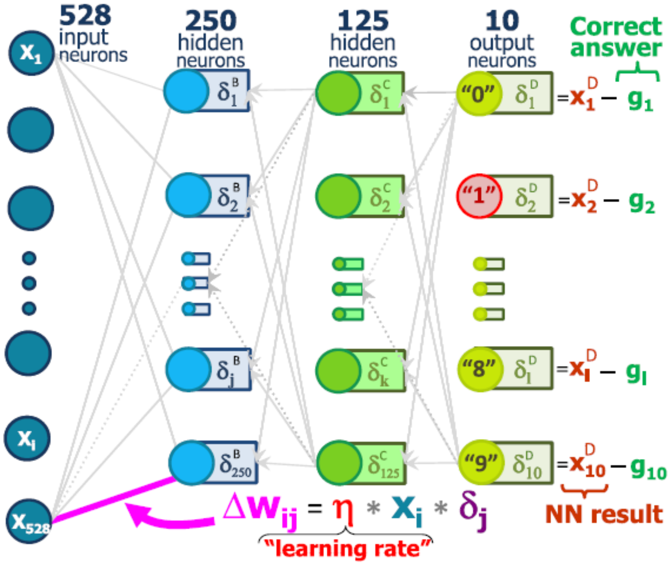

Fig. 5. Sketch of an ANN depicting the supervised learning process by the backpropagation scheme. The error from Eq. (3) is back-propagated from output to input for weight update according to Eq. (4). Reproduced with permission from [29]. Copyright IEEE (2015).

2.2 Weight update of resistive switching devices

The weight update according to Eq. (4) poses a significant challenge in the design of PCM and RRAM synapses, in that both potentiation and depression must be achieved via incremental steps. However, in general, only one of the 2 operations can be gradually achieved: for instance, the crystallization process in PCM is incremental in that the application of sequential pulses causes more amorphous phase to change the crystalline state [21, 24]. On the other hand, amorphization process is only a function of the applied voltage and current in each pulse, without any significant dependence on the actual state of the PCM device. As a result, only incremental potentiation can be achieved in PCM devices, *i.e.*, Eq. (4) is only applicable for increasing the synaptic weight w_{ij}, or PCM conductance G_{ij}. Conversely, RRAM devices generally show incremental depression by repeating reset pulses [56], while incremental potentiation requires specific device materials, such as interfacial switching $Pr_{1-x}Ca_xMnO_3$ [57] or bilayer-stacked TaO_x-TiO_2 [58]. Another issue with resistive arrays is that a conductance G_{ij} can only map a positive weight w_{ij}, whereas both positive and negative weights are generally needed to represent input patterns in ANNs [55, 57].

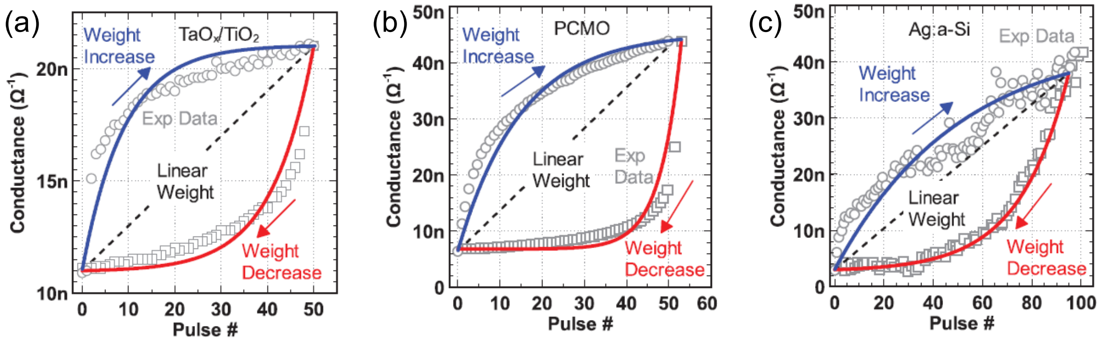

Fig. 6. Measured update characteristics for weight increase (potentiation) and weight decrease (depression). Data show the measured conductance in response to the application of a sequence of pulses with constant positive/negative amplitude. Analog potentiation/depression is needed for backpropagation in the supervised learning of ANNs. Reproduced with permission from [59]. Copyright IEEE (2015).

These problems can be overcome by a differential approach, where each synapse is represented by 2 memory elements, *e.g.*, 2 PCM devices [28, 29] or 2 RRAM devices [57], and synaptic currents are obtained as the difference between the 2 paths in the differential scheme, namely:

$$I_j = \sum_i \left(G_{ij}^{+} - G_{ij}^{-}\right) V_i \qquad (6)$$

where G_{ij}^{+} and G_{ij}^{-} are the conductance values of the positive and negative resistive elements in the synaptic cell at position (i,j) in the array [57]. As a result, potentiation of weight w_{ij} can be achieved by either potentiation of conductance G_{ij}^{+}, or depression of conductance G_{ij}^{-}, depending on potentiation or depression being incremental in the adopted resistive switching synapse.

Another key issue for resistive switching synapse is the linearity and symmetry of the plasticity characteristics. Clearly, update according to Eq. (4) requires that the amount of potentiation/depression depends only on the number of pulses applied to the synapse element, not on the actual weight of the synapse which is generally not known. As a result, an ideal synapse should be linear, in that a certain amount of potentiation/depression can be achieved by a fixed amount of set/reset pulses irrespective of the resistive state of the device. On the other hand, real devices fail in satisfying linearity as shown by the weight update characteristics in Fig. 6 [59]: all the 3 reported cases show non-linear change of conductance along both the potentiation and depression branches of the characteristics. For instance, potentiation is relatively steep if potentiating pulses are applied to a device with low conductance, whereas slow potentiation is obtained for relatively high conductance. Similarly, steep depression occurs for high conductance states, with saturation taking place at low conductance. As a figure of merit, one can define a linearity factor α, which controls the power dependence of the normalized weight *g* on the number of pulses given by:

$$g = \frac{G - G_0}{G_1 - G_0}, \tag{7}$$

where *g* varies between 0 and 1 as the synaptic weight *G* changes from the lowest value G_0, corresponding to the HRS, and the highest value G_1, corresponding to the LRS [57]. The normalized weight increases with the normalized number of pulses x, also changing between 0 and 1, according to the power law:

$$g = x^{\frac{1}{\alpha}}, \tag{8}$$

with the non-linearity factor α generally varying from 3 to 6 for potentiation and being equal to 1 only for ideally linear potentiation/depression synapses.

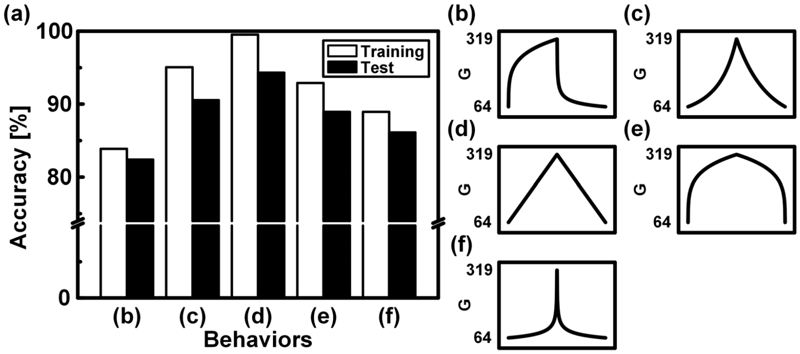

Fig. 7. (a) Learning efficiency for training, namely the probability for correct recognition of a pattern belonging to the data set and already submitted to the network during training, and for test, namely the probability of correct recognition of a pattern which was not submitted to the network during training. The learning efficiency was studied for various types of update characteristics, such as non-linear/non-symmetric characteristics with α > 1 for potentiation and α < 1 for depression (b), non-linear/symmetric

characteristics with $\alpha < 1$ relatively close to 1 (c), linear characteristics (d), non-linear/symmetric characteristics with large $\alpha > 1$ (e), and non-linear/symmetric characteristics with small $\alpha < 1$ (f). Reproduced with permission from [57]. Copyright IEEE (2015).

The lack of linear update causes inefficient learning in Fig. 7, showing the learning efficiency for training, *i.e.*, the probability for correct recognition of a pattern belonging to the data set and already submitted to the network during training, and for test, *i.e.*, the probability of correct recognition of a pattern which has not been submitted to the network during training [57]. The learning efficiency has been studied for various types of update characteristics, such as non-linear/non-symmetric characteristics with $\alpha > 1$ for potentiation and $\alpha < 1$ for depression (b), non-linear/symmetric characteristics with $\alpha < 1$ relatively close to 1 (c), linear characteristics (d), non-linear/symmetric characteristics with large $\alpha > 1$ (e), and non-linear/symmetric characteristics with small $\alpha < 1$ (f). In general, the learning efficiency suffers from both non-linearity and asymmetry, with the best performance for linear characteristics (d), and the worst behavior for non-linear, non-symmetric update (b). These results support the importance of a broad research scope aiming at linearity and symmetry in update characteristics, either by material engineering [57-59], or by circuit design [60]. Another important device property is the available dynamic range, namely the ratio between the highest and the lowest achievable conductance. A higher dynamic range allows to accommodate a larger number of intermediate steps, thus providing improved ANN performances. Toward this goal, an interesting approach is the periodic carry concept [61], where, instead of using one device for G^+ and one for G^-, more devices are employed, each of them with a varying significance obtained by weighting the single device contributions with incremental coefficients (*e.g.*, 1, 1/5, 1/25 and 1/125 [61]). With the periodic carry concept, the available number of levels can thus increase linearly with the number of devices employed.

2.3 Acceleration of ANNs with resistive switching devices

Crossbar architectures display large potential advantages for the calculation of MVM operations, thus being ideal to implement FC-ANNs. Several studies have been proposed to either perform training [29, 30] or directly program pre-trained weights [62, 63]. The goal of the crossbar implementation is to achieve substantial speedup over conventional approaches to training and forward inference of deep learning networks, generally adopting the graphical processing unit (GPU) for fast MVM [29]. While GPUs show high performance in training convolutional neural networks (CNNs) [64], due to their ability to rapidly manipulate and move kernel weights during convolution operations, the performance on FC-ANNs is reduced because of the very large amount of weights to be trained. In this context, dense crossbar arrays of resistive devices represent an optimal solution, ideally reducing the duration of the weight update within an entire crossbar, hence ANN layer, to a single clock step [29].

To combine the fast training in crosspoint architectures with the flexibility of GPUs, recently proposed analog/digital hybrid systems implement the crossbar array as an analog computational unit able to largely accelerate the calculation of the MVM [65-68], thus allowing to relax the computational load on the digital section [68]. For instance, a dot-product engine was proposed where the resistive crossbar array efficiently calculates MVM via vector scalar product $\mathbf{a}\cdot\mathbf{b} = \Sigma a_i b_i$, achieving in simulation software-like accuracy on MNIST digit recognition by implementing pre-trained weights and efficiently performing forward inference [65, 66]. In a recently-proposed hybrid scheme, ANN training is performed by storing the weight updates into a high-precision memory, and then transferring the cumulated weight change ΔW on a crossbar memory only when ΔW shows a size comparable with the device conductance change [68]. Based on these reports, hybrid systems incorporating analog/digital CMOS circuits and crosspoint arrays of emerging memories, such as PCM and RRAM, appear as the most promising approach to ANNs and other neural networks for deep learning applications.

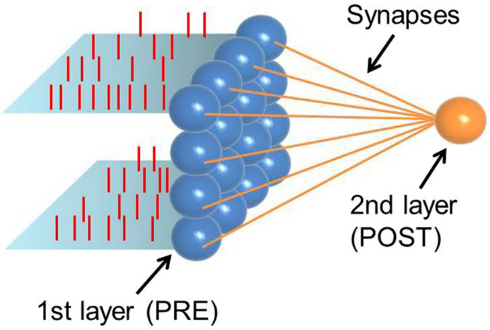

Fig. 8. Sketch of a feedforward FC SNNs with a single-layer perceptron architecture. Reproduced from [44].

3. Spiking neural networks for unsupervised learning

In a typical ANN, the input and output information are carried by the amplitude of a voltage or a current, which are linked to the synaptic weights by Eq. (1). While input/output signals might be either continuous or pulsed, there is no information carried by the time variables of the pulse, such as pulse width, or the frequency and the time of the occurrence of a stimulus. This is opposite to how the brain represents information, which is mapped by the specific neuron being active, and by the specific time of the neuron action, or spike. Such a spatiotemporal coding is what makes the brain highly energy efficient and highly functional in representing and elaborating complex information [69-71]. Another conjectured mode of information coding in the brain is rate coding [69-71], where the average rate of neuron spiking is used to describe the relevant input/output information.

Neuromorphic systems aiming at replicating the correct data processing in the brain usually adopt the spiking neural network (SNN) architecture with suitable spatio-temporal or rate coding of the information. Fig. 8 illustrates a typical SNN with single-layer perceptron structure, consisting of a first layer of neurons, each referred to as a pre-synaptic neuron (PRE), and a second layer with a single post-synaptic neuron (POST), which is connected to each PRE by a synapse [44]. The PRE spikes are transmitted through the synaptic connections to the POST for processing and learning.

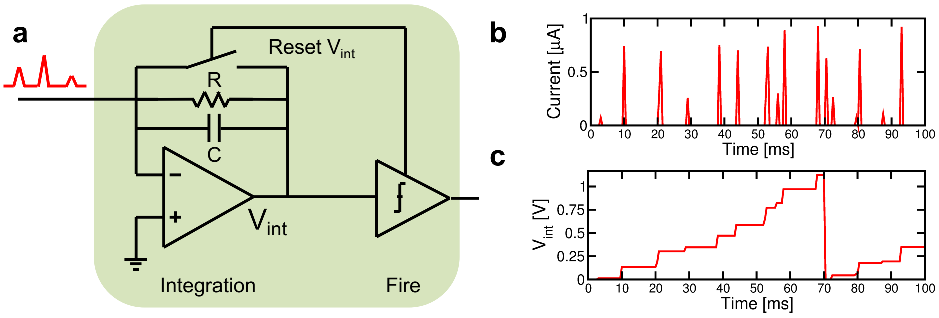

Fig. 9. (a) Sketch of an integrate&fire neuron, (b) input spikes and (c) corresponding internal potential V_{int} according to simulations. At fire, the accumulation is reset and a spike is generated toward the next layer of neurons. Reproduced from [44].

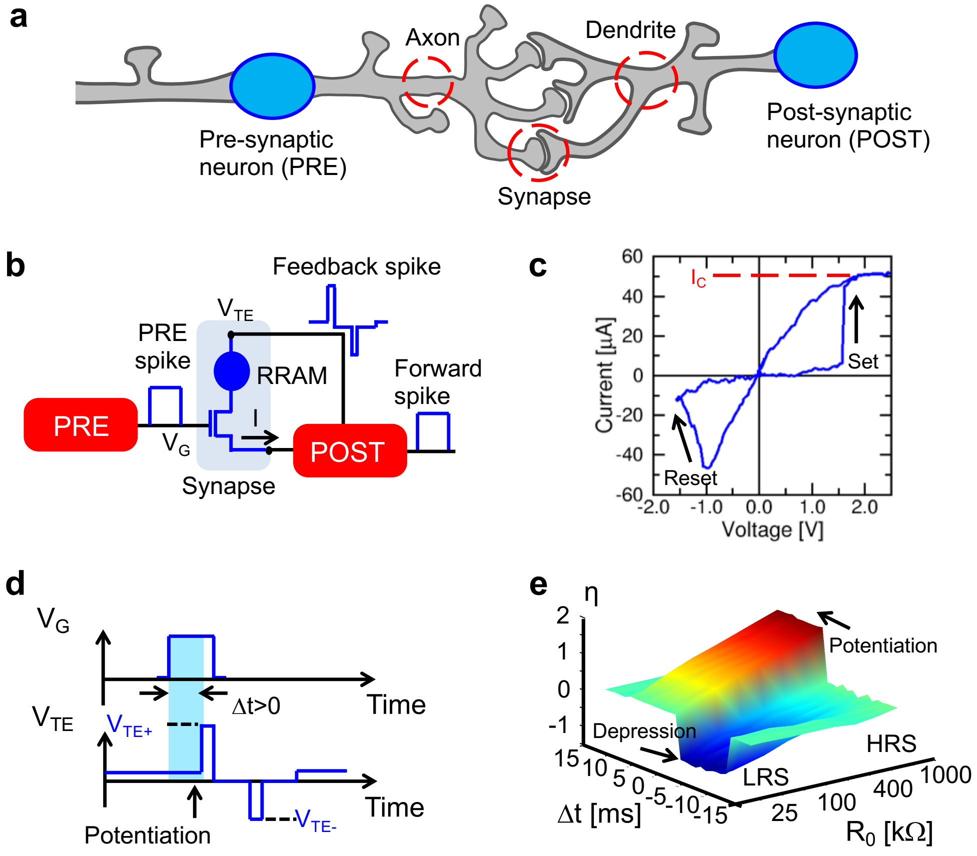

Fig. 10. (a) Sketch of a biological system of 2 neurons connected by a synapse, (b) synapse circuit implementation into a hybrid CMOS/RRAM 1T1R structure, (c) I-V curve of the 1T1R HfO_2-based RRAM with bipolar switching characteristics, (d) PRE and POST spikes overlapping for Δt > 0, and (e) measured STDP characteristics of η as a function of spike time delay and initial synaptic resistance R_0. Reproduced from [44].

3.1 Neurons and synapses in SNNs

In a SNN, neurons typically have a leaky integrate&fire (LIF) behavior, where input spikes are accumulated into an internal variable, usually referred to as membrane potential or internal potential V_{int}. As V_{int} reaches a threshold value, the neuron fires, *i.e.*, emits a spike toward the next layer of neurons, while resetting V_{int} back to zero. Fig. 9a shows the typical circuit of an electronic LIF neuron [44], including a first stage for leaky integration of an input current, with output V_{int}. The latter serves as input for the second 'fire' stage, consisting of a voltage-controlled astable circuit for generating a spike as V_{int} overcomes a given threshold. At the fire event, the fire stage also sends a control signal back to the integration to reset the internal voltage V_{int} to zero by short-circuiting the feedback capacitor C. Fig. 9b shows a typical spiking current signal, while Fig. 9c shows the calculated V_{int} from a circuit simulation with the fire threshold V_{th} = 1 V [44].

Fig. 10a illustrates a biological system of 2 neurons connected by a synapse. To replicate such elementary system in silico, the synapse can be implemented by a hybrid CMOS/RRAM structure as represented in Fig. 10b. Here, the synapse combines a RRAM element with a field-effect transistor (FET) in a 1-transistor/1-resistor (1T1R) structure, where the HfO_2-based RRAM shows a bipolar switching behavior with a set transition from high resistance state (HRS) to low resistance state (LRS) at positive voltage, and a reset transition from HRS to LRS for negative voltage (Fig. 10c). The current during set transition is forced to remain below a compliance current I_C, which can be controlled by the gate voltage of the series FET. When the PRE emits a spiking voltage to the gate of the FET, a small current proportional to the 1T1R conductance flows thanks to the constant voltage V_{TE} applied to the top electrode. The bottom electrode is connected to the input node of the POST, which thus integrates all spiking currents from the connected PREs. When the internal (membrane) potential V_{int} of the POST reaches a threshold, the POST fires, which consists of the generation of a feedforward spike to the next layer of neurons. Also, the POST generates a *feedback* spike composed by a positive voltage pulse and a negative voltage pulse, which can temporally overlap with the PRE spike. In case the PRE spike is temporally preceding the POST spike by a time delay $\Delta t > 0$ (Fig. 10d), the positive branch of the feedback spike partially overlaps with the PRE spike, thus causing a set transition of the RRAM, or synaptic potentiation. Conversely, if the PRE spike is temporally following the POST spike by $\Delta t < 0$, the negative branch of the feedback spike partially overlaps with the PRE spike, thus inducing a reset transition of the RRAM, or synaptic depression. The delay-dependent RRAM conductance changes reproduce STDP, where potentiation and depression occur for positive and negative Δt, respectively [72-76]. The PRE spike is 10 ms long, which is designed to cause no impact on the 1T1R conductance for relatively long delays, *i.e.*, $|\Delta t| > 10$ ms, according to the STDP learning rule. Fig. 10e shows the experimental STDP curve, reporting the ratio $\eta = \log_{10}(R_0/R)$ as a function of the delay time Δt for various initial resistances R_0 of the RRAM device. Data indicate depression and potentiation for $\Delta t < 0$ and $\Delta t > 0$, respectively [44]. Note that the 1T1R synapse is bistable, as a weight update always leads to a full set transition to LRS, or a full reset transition to HRS, irrespective of the initial resistance R_0. This implementation of STDP contrasts with the ANN trained with the backpropagation algorithm in Sec. 2, as it only requires two synaptic levels, thus being much more resilient to RRAM variability [77] and noise [78, 79]. On the other hand, the repeated request for full transitions may cause a stronger device degradation, with respect to the incremental conductance changes in backpropagation-based ANN.

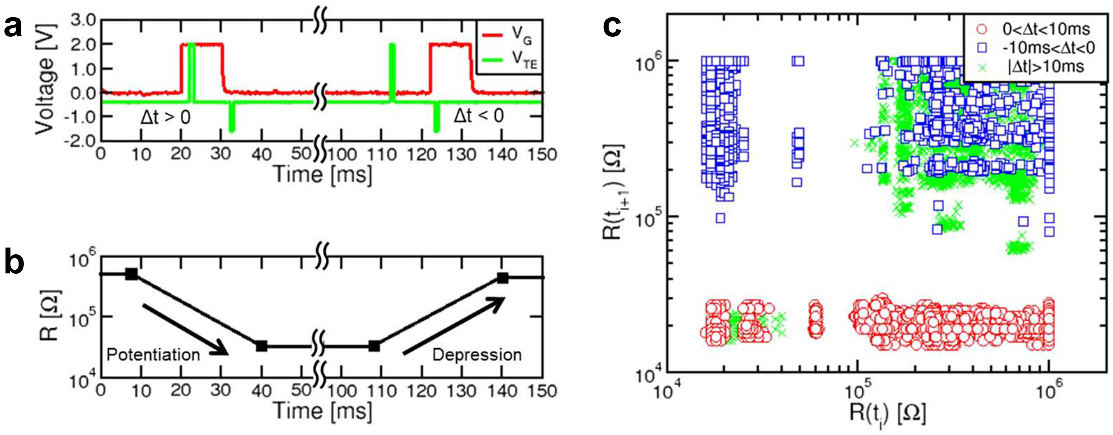

Fig. 11. (a) Measured PRE and POST spikes in a hardware 4x4 feedforward SNN, (b) corresponding change of resistance indicating potentiation (Δt = +3 ms) and depression (Δt = -7 ms), and (c) correlation plot of final resistance R(t_{i+1}) and initial resistance R(t_i) for various delays in the STDP synapse, leading to potentiation (positive Δt), depression (negative Δt), and no change (large Δt). Reproduced from [44].

3.2 Unsupervised learning by STDP

A key difference between brain-inspired SNNs and ANNs for deep learning is the type of learning taking place at synaptic level. Since there is no direct supervision in the brain, SNNs usually implement unsupervised learning by STDP, where the synaptic network autonomously learns a pattern that is stochastically submitted to the network.

Unsupervised learning by STDP has been demonstrated in hardware SNNs with 1T1R synapses based on feedforward structure shown in Fig. 8. Fig. 11 shows the experimental STDP behavior for a 1T1R synapse with the structure of Fig. 10, which was implemented in a 4x4 feedforward perceptron [44]. Based on the temporal overlap between the PRE and POST spikes in the 1T1R synapse (Fig. 11a), the synaptic weight undergoes potentiation for Δt > 0 (Δt = +3 ms in Fig. 11b) and depression for Δt < 0 (Δt = -7 ms in Fig. 11b). Fig. 11c summarizes the STDP response of the synapse, showing the correlation between the resistance $R(t_i)$ measured before the application of the spikes, and the resistance $R(t_{i+1})$ measured after the application of the spikes, for various pulse combinations. Cases with 0 < Δt < 10 ms reveal clear transitions to the LRS, while the synapse switches to the HRS for -10 ms < Δt < 0. Finally, cases for |Δt| > 10 ms show no resistance variation, as the pulses are never applied to the device at the same time [44].

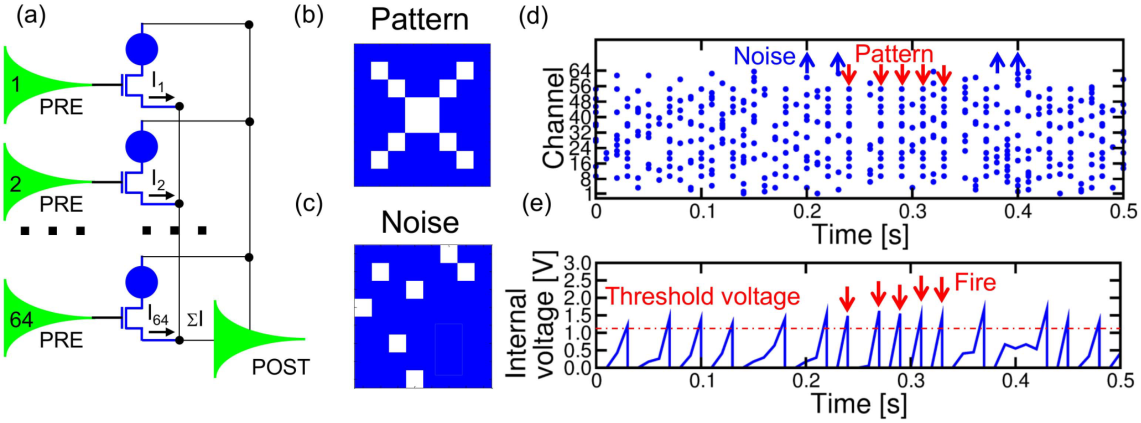

Fig. 12. (a) Sketch of a feedforward SNN with 1T1R synapses, (b) pattern and (c) noise adopted for unsupervised learning, (d) input spikes and (e) corresponding V_{int} indicating fire in response to the presentation of the pattern after learning. Reproduced with permission from [40]. Copyright IEEE (2016).

Fig. 12a shows the circuit implementation of perceptron network (Fig. 8), evidencing an architecture where many PREs are connected to the POST by 1T1R synapses. To achieve unsupervised specialization on an input pattern, the pattern (*e.g.*, the “X” in Fig. 12b) is repeatedly submitted by the PREs through the synaptic channels, and on purpose randomly alternated with noise (Fig. 12c), as indicated by the raster plot in Fig. 12d, which shows the pixels of the 64 channels as a function of time. Thanks to the unsupervised learning, once the network has specialized on the input pattern, the internal voltage V_{int} in Fig. 12e shows a clear correlation between fire events and input pattern [40].

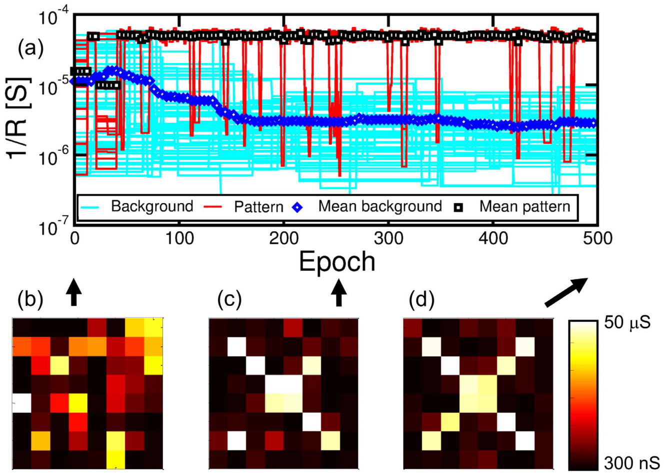

Fig. 13. (a) Measured conductance of the 1T1R synapses during unsupervised learning, and color plots of the synaptic weights for (b) initial state, (c) after 250 epochs, and (d) after 500 epochs. While potentiation of pattern synapses is almost immediate, the depression of background synapses is more gradual due to the uncorrelated, low-density noise spikes. Reproduced with permission from [40]. Copyright IEEE (2016)

Fig. 13a shows the simulated behavior of the synaptic weights during the unsupervised training described in Fig. 12. Simulations were carried out according to a stochastic model of RRAM devices, where set/reset events led to a statistically distributed resistance values replicating the experimentally observed distributions [40]. Starting from a random distribution of synaptic conductances, the synapses within the pattern channels converge to full LRS in around 50 epochs, while synapses in the other channels, also known as background synapses, show a gradual depression, as evidenced by synaptic conductance maps in Fig. 13b-d. The contrast between the abrupt potentiation and the gradual depression origins from the different roles of input pattern and noise. In fact, the pattern submission generates an immediate potentiation of pattern synapses, because pattern PRE spikes are correlated in time, thus are followed by a POST fire which causes potentiation according to the STDP rule. On the other hand, PRE noise spikes are not correlated, therefore are much more likely to follow a POST fire, rather than anticipate a POST fire. Synapse depression thus occurs according to the STDP rule. However, since noise spiking density is relatively small compared to the pattern density, the depression rate is lower than the potentiation rate, the latter approaching the one- or few-shot learning speed [80]. Higher noise density leads to a faster background depression, hence increased overall training speed. However, excessive noise prevents the initial pattern potentiation, due to the competition between pattern and noise in inducing POST fire. The maximum amount of injected noise, hence maximum learning rate, is thus dictated by a trade-off between learning speed and training robustness and stability [81].

3.3 Hardware demonstration of unsupervised learning

A FC SNN with perceptron architecture was experimentally demonstrated by implementing PRE/POST neurons and synapses on a printed circuit board (PCB) [44]. An Arduino Due microcontroller (μC) was adopted to describe the spiking neurons, while the 1T1R HfO_2 RRAM synapses were wired together to build a 4x4 perceptron network.

Fig. 14a shows the schematic circuit layout, while Fig. 14b shows the practical hardware implementation on the PCB [44].

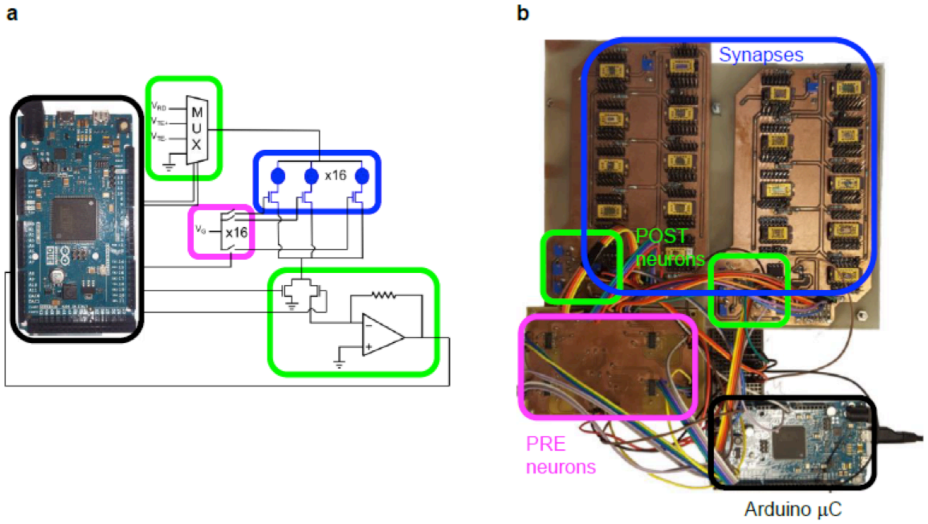

Fig. 14. (a) Sketch and (b) picture of the hardware SNN implemented on a PCB. The 1T1R synapses and the PRE/POST neurons controlled by the microcontroller are indicated. Reproduced from [44].

Fig. 15 shows the measured synaptic weights during the unsupervised learning by stochastic spikes. The visual pattern consisted of a diagonal from lower-left to upper-right of the 4x4 image and was alternated with random noise functional for depression. Starting from a random distribution of synaptic conductance, synapses automatically adapt to the submitted pattern, reaching a complete pattern potentiation and background depression in about 1000 epochs, as shown by the color plot of the synaptic weights in Fig. 15a-d. Fig. 15e shows the spiking activity alternating pattern and noise, while Fig. 15f shows the detailed measured conductance evolution with time. The noise density was relatively low (about 5%), due to the small size of the visual pattern. Despite the programming variability and resistance fluctuations which characterize the RRAM, Fig. 15f indicates a marked resistance window between the pattern synapses at LRS and background synapses at HRS, thus supporting the robustness of the learning algorithm. In addition to pattern specialization and recognition, one of the advantages of unsupervised algorithm is the ability to adapt to input variations, at least in case input variations are slower than the learning time. This adaptive behavior contrasts with ANNs trained with backpropagation algorithm, where the "catastrophic forgetting" [82] prevents the network to adapt to modifications of the input information, such as the addition of a new class, thus forcing to retrain the entire network rather than just adapt a portion of the weights.

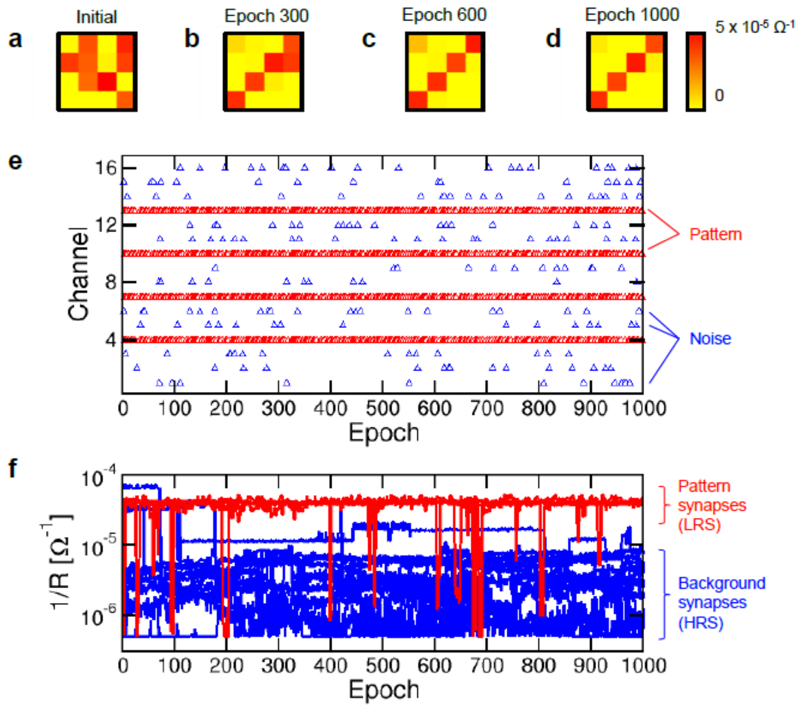

Fig. 15. Color plot of synaptic conductance values for (a) initial states, (b) after 300 epochs, (c) after 600 epochs, and (d) after 1000 epochs, (e) corresponding input spikes

and (f) real-time behavior of the synaptic conductance. Pattern synapses are potentiated to LRS, while background synapses are depressed to HRS. Reproduced from [44].

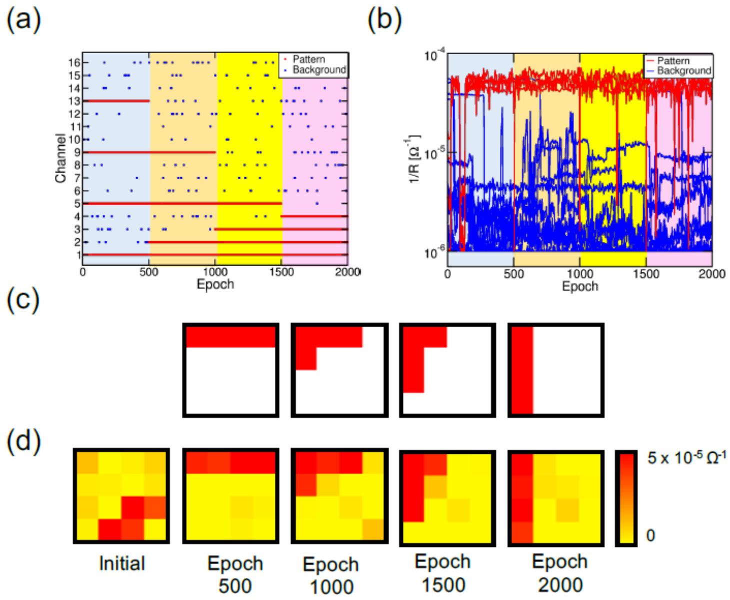

Fig. 16. (a) Input spikes, (b) synaptic weight evolution with time, (c) submitted patterns for each phase and (d) corresponding weights for the initial state and after each training phase. The submitted pattern was initially a top bar, then shifted to the left by one pixel at a time at every epoch. Reproduced from [81].

To demonstrate the adaptability of the STDP algorithm, Fig. 16 shows the successive learning of four different patterns, which were submitted during presentation phases at increasing times. During each phase, a different pattern was applied (Fig. 16a), leading to the adaptation of the synaptic weights to the submitted pattern (Fig. 16b). Fig. 16c shows the four visual patterns, consisting of 4 pixels gradually shifting from the top bar to the left bar in three steps. Fig. 16d shows the color plot of the synaptic weights, revealing a fast adaptability to the variation of the input pattern. Interestingly, the synapses not involved in the change of the input pattern do not significantly change their conductance after input shift, thus confirming the robustness of STDP to input variations [81].

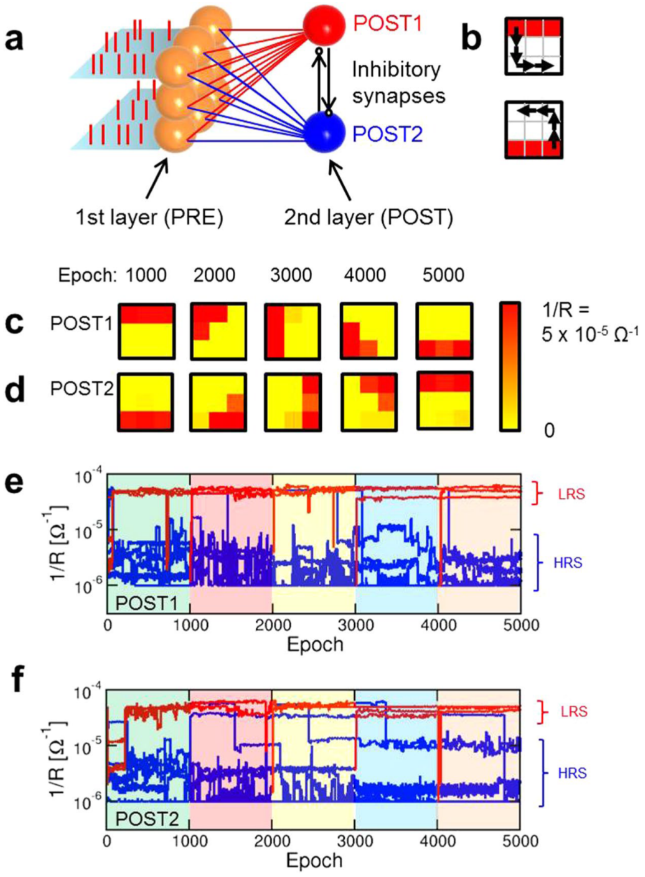

Fig. 17. (a) SNN for multi-pattern learning by POST1 and POST2, (b) submitted pattern 1 (top bar with anticlockwise shift) and pattern 2 (bottom bar with anticlockwise shift), color plot of the synaptic conductance for synapses pointing to (c) POST1 and (d) POST2, measured synaptic weights for (e) POST1 and (f) POST2 as a function of epoch. Reproduced from [44].

STDP networks can also enable the classification and recognition of more than one pattern by implementing inhibitory synapses. Fig. 17a shows a perceptron SNN where the input layer is fully connected to two POST neurons, defined as POST1 and POST2. The two POSTs are mutually connected by inhibitory synapses, namely non-adaptive synapses which provide a negative current spike from a firing POST to the other. For instance, when POST1 fires in response to submitted pattern 1, POST2 receives a negative spike, thus inducing a decrease of the internal potential and preventing POST2 to fire in response to the same pattern. This allows each neuron to specialize on separate patterns, thus maximizing the information storage and recognition capability of the network. Two 3x3 RRAM synaptic arrays were physically implemented to fully connect the 9 input axon channels to POST1 and POST2. To demonstrate learning and synaptic adaptation in the network, the two moving patterns in Fig. 17b, consisting of top/bottom horizontal stripes shifting to the clockwise or anticlockwise direction by one pixel every submission phase, were submitted. The patterns were randomly presented one at a time at the PRE channels, and were alternated with noise according to the previously discussed stochastic approach. Every 1000 epochs the patterns were shifted as indicated by arrows. Fig. 17c and d show the color plots of the measured synaptic weights for POST1 and POST2, respectively, at the end of each training phase. Note that each POST randomly learns one of the 2 patterns initially, then remains locked to the same pattern during the subsequent phase, due to the tendency to minimize the synaptic weight change from one phase to the other. Fig. 17e and f show the measured weight conductance for increasing epochs during each phase for POST1 and POST2, respectively. Pattern synapses are almost immediately potentiated to enable specialization of POST1 and POST2 to learn different patterns and track them during subsequent phases, thus keeping memory of the previous stored pattern.

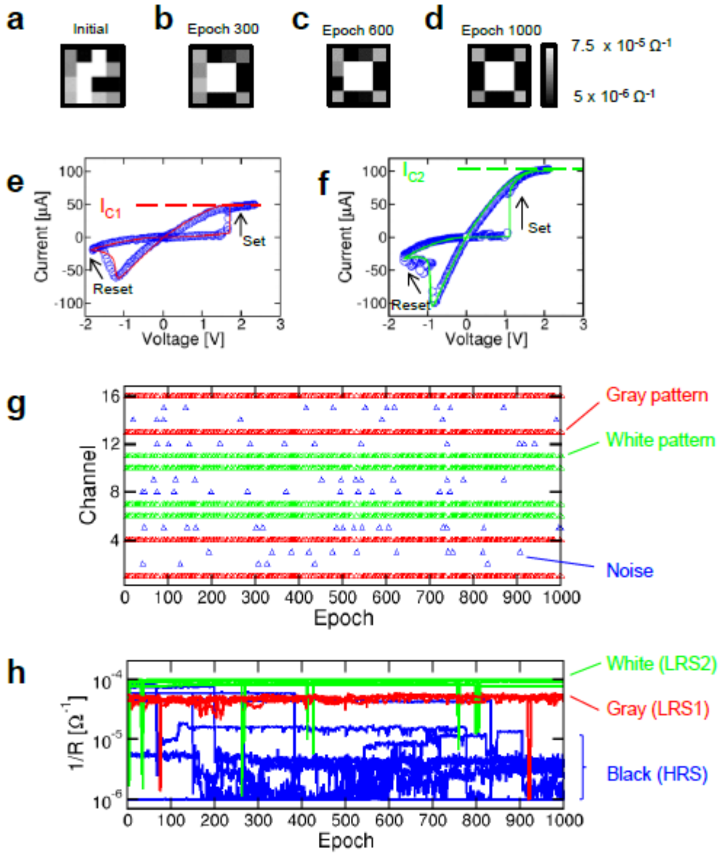

Fig. 18. Synaptic weights in a gray-scale color plot for (a) initial states, (b) after 300 epochs, (c) after 600 epochs, (d) after 1000 epochs, (e) measured I-V curve of the HfO_2 RRAM for a low compliance current I_{C1} = 50 μA, corresponding to gray color, and (f) I_{C2} = 100 μA corresponding to white color, (g) input spikes and (h) measured synaptic weights as a function of time. Reproduced from [44].

Bistability of 1T1R synapses allows for robust learning thus providing strong advantages in a real implementation, however the amount of information that can be encoded is only one bit, while several cases need more capacity. For instance, gray-scale images require a larger amount of information during synapse training, namely, not only the position of the pattern, but also the signal amplitude at that position. Fig. 18 shows an experimental demonstration of gray-scale image recognition, where the pattern was characterized by white and gray pixels. Fig. 18a-d shows the conductance color plots of experimental RRAMs during pattern learning. Information was stored into the 1T1R synapses changing the compliance current I_c proportional to pixel brightness, where gray correspond to I_c =50 μA (Fig. 18e), while white corresponds to the highest I_c =100 μA (Fig. 18f). The black background was achieved by noise-induced depression. Fig. 18g shows the raster plot of input spikes, while Fig. 18h shows the measured synaptic conductances for increasing epochs, supporting gray-scale learning with gray and white patterns reaching separate LRS conductance values LRS1 and LRS2, respectively [44].

4. Conclusions and outlook

This chapter reviews the state of the art regarding neuromorphic computing using resistive switching memory (RRAM) devices. Two computing approaches have been covered, namely the ANN for supervised training and the SNN for brain-inspired unsupervised learning via STDP. ANNs find application in computer vision and can improve the training speed and energy efficiency with respect to CMOS-based systems, such as the GPU, thanks to physical computation of MVM within the memory array. The main challenge is the control of conductance in the RRAM, due to variability and non-linearity of weight update characteristics. On the other hand, RRAM seems a promising solution for synaptic elements in brain-inspired SNNs, due to robust learning and relatively-easy implementation of bio-realistic learning schemes, such as STDP and spike-rate dependent plasticity (SRDP) rules. Unsupervised learning via STDP in SNNs was demonstrated both in simulation and hardware experiments, thus supporting the feasibility of RRAM-based SNNs for brain-inspired neuro-computing.

Although RRAM seems promising for synaptic elements in neural networks, there are still several challenges to reach commercial viability with respect to other existing technologies. In ANN applications, RRAM synapses are still affected by relatively slow programming compared to SRAM and DRAM, strong non-linearity of weight update [57-60], and variability issues [31,77]. In particular, gradual set and reset processes are usually observed in a limited class of RRAM devices, which still requires optimization to improve the symmetry and linearity and thus achieve robust learning in supervised training. Materials engineering, *e.g.*, adopting interface-type switching in bilayer structures [58], or synaptic circuit engineering [60] are needed to improve and optimize RRAM characteristics for analog ANNs.

Regarding brain-inspired SNNs, the application scenario for this technology is not clear yet. The implementation of cognitive primitives similar to the human brain is extremely promising for realizing image and speech recognition in hardware, with relatively high energy efficiency and large information storage density. However, these applications still require synaptic devices capable of closely resembling the biological learning rules in the brain, such as short-term activation in synaptic plasticity, which might enable bio-realistic learning by the physics of the device [83]. The architecture to achieve this goal might require both feedforward and recurrent SNNs, such as Hopfield networks with attractor-based dynamics [84]. A broadly interdisciplinary approach involving device engineering, circuit design, and neuroscience is thus needed to create a research roadmap toward the development of truly brain-inspired neuromorphic computing.

5. Acknowledgments

This chapter has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 648635).

References

[1] G.E. Moore, Cramming more components onto integrated circuits, Electronics (1965) 114-117.

[2] K.J. Kuhn, Considerations for ultimate CMOS scaling, IEEE Trans. Electron Devices 59 (7) 1813-1828 (2012). DOI: 10.1109/TED.2012.2193129

[3] A.M. Ionescu and H. Riel, Tunnel field-effect transistors as energy-efficient electronic switches, Nature 479 (7373) (2011) 329-337. DOI: 10.1038/nature10679

[4] S. Salahuddin and S. Datta, Use of negative capacitance to provide voltage amplification for low power nanoscale devices, Nano Lett. 8 (2) (2008) 405-410. DOI: 10.1021/nl071804g

[5] D.E. Nikonov and I.A. Young, Overview of beyond-CMOS devices and a uniform methodology for their benchmarking, Proc. IEEE 101 (12) (2013) 2498-2533. DOI: 10.1109/JPROC.2013.2252317

[6] 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, and D.S. Modha, A million spiking-neuron integrated circuit with a scalable communication network and interface, Science 345 (6197) (2014) 668-673. DOI: 10.1126/science.1254642

[7] G. Indiveri and S.-C. Liu, Memory and information processing in neuromorphic systems, Proc. IEEE 103 (8) (2015) 1379-1397. DOI: 10.1109/JPROC.2015.2444094

[8] R. Waser and M. Aono, Nanoionics-based resistive switching memories, Nat. Mater. 6 (2007) 833-840. DOI: 10.1038/nmat2023

[9] 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, 100 (6) (2012) 1951-1970. DOI: 10.1109/JPROC.2012.2190369

[10] D. Ielmini, Resistive switching memories based on metal oxides: Mechanisms, reliability and scaling, Semicond. Sci. Technol. 31 (2016), 063002. DOI: 10.1088/0268-1242/31/6/063002

[11] M. Wuttig and N. Yamada, Phase-change materials for rewriteable data storage, Nat. Mater. 6 (2007) 824-832. DOI: 10.1038/nmat2009

[12] S. Raoux, W. Welnic and D. Ielmini, Phase change materials and their application to non-volatile memories, Chem. Rev. **110** (1) (2010) 240-267. DOI: 10.1021/cr900040x

[13] G.W. Burr, M.J. Breitwisch, M. Franceschini, D. Garetto, K. Gopalakrishnan, B. Jackson, B. Kurdi, C. Lam, L.A. Lastras, A. Padilla, B. Rajendran, S. Raoux, and R.S.

Shenoy, Phase Change Memory Technology, J. Vac. Sci. Technol. B 28 (2) (2010) 223-262. Online: http://doi.org/10.1116/1.3301579

[14] C. Chappert, A. Fert, and F.N. Van Dau, The emergence of spin electronics in data storage, Nat. Mater. 6 (2007) 813-823. DOI: 10.1038/nmat2024

[15] B. Dieny, R.C. Sousa, J. Hérault, C. Papusoi, G. Prenat, U. Ebels, D. Houssameddine, B. Rodmacq, S. Auffret and L.D. Buda-Prejbeanu, Spin-transfer effect and its use in spintronic components, Int. J. Nanotechnol. **7**, 591 (2010). DOI: 10.1504/IJNT.2010.031735

[16] A.D. Kent and D.C. Worledge, A new spin on magnetic memories, Nat. Nanotechnol. 10 (2015) 187-191. DOI: 10.1038/nnano.2015.24

[17] J. Borghetti, G.S. Snider, P.J. Kuekes, J.J. Yang, D.R. Stewart and R.S. Williams, 'Memristive' switches enable 'stateful' logic operations via material implication, Nature 464 (2010) 873-876. DOI: 10.1038/nature08940

[18] S. Balatti, S. Ambrogio, and D. Ielmini, Normally-off logic based on resistive switches - Part I: Logic gates, IEEE Trans. Electron Devices 62 (6) (2015) 1831-1838. DOI: 10.1109/TED.2015.2422999

[19] B. Chen, F. Cai, J. Zhou, W. Ma, P. Sheridan, and W.D. Lu, Efficient in-memory computing architecture based on crossbar arrays, in: IEDM Tech. Dig 2015, pp. 17.5.1-17.5.4. DOI: 10.1109/IEDM.2015.7409720

[20] P. Huang, J. Kang, Y. Zhao, S. Chen, R. Han, Z. Zhou, Z. Chen, W. Ma, M. Li, L. Liu, and X. Liu, Reconfigurable nonvolatile logic operations in resistance switching crossbar array for large-scale circuits, Adv. Mater. 28 (44) (2016) 9758-9764. DOI: 10.1002/adma.201602418

[21] M. Cassinerio, N. Ciocchini and D. Ielmini, Logic computation in phase change materials by threshold and memory switching, Adv. Mater. 25 (41) (2013) 5975-5980. DOI: 10.1002/adma.201301940

[22] Y. Li, Y.P. Zhong, Y.F. Deng, Y.X. Zhou, L. Xu, and X.S. Miao, Nonvolatile "AND," "OR," and "NOT" Boolean logic gates based on phase-change memory, J. Appl. Phys. 114 (2013) 234503. DOI: 10.1063/1.4852995

[23] D. Loke, J.M. Skelton, W.-J. Wang, T.-H. Lee, R. Zhao, T.-C. Chong, and S.R. Elliott, Ultrafast phase-change logic device driven by melting processes, Proc. Natl. Acad. Sci. USA (PNAS) 111 (2014) 13272-13277. DOI: 10.1073/pnas.1407633111

[24] C.D. Wright, Y. Liu, K.I. Kohary, M.M. Aziz, and R.J. Hicken, Arithmetic and biologically-inspired computing using phase-change materials, Adv. Mater. 23 (30) (2011) 3408-3413. DOI: 10.1002/adma.201101060

[25] P. Hosseini, A. Sebastian, N. Papandreou, C.D. Wright, and H. Bhaskaran, Accumulation-based computing using phase-change memories with FET access devices, IEEE Electron Device Lett. 36 (9) (2015) 975-977. DOI: 10.1109/LED.2015.2457243

[26] S.N. Truong, and K.-S. Min, New memristor-based crossbar array architecture with 50% area reduction and 48% power saving for matrix-vector multiplication of analog

neuromorphic computing, J. Semicond. Technol. Sci. 14 (3) (2014) 356-363. DOI: 10.5573/JSTS.2014.14.3.356

[27] P. Gu, B. Li, T. Tang, S. Yu, Y. Cao, Y. Wang, H. Yang, Technological exploration of RRAM crossbar array for matrix-vector multiplication, in: 2015 20th Asia and South Pacific Design Automation Conference (ASP-DAC) 2015, pp. 1-6. DOI: 10.1109/ASPDAC.2015.7058989

[28] 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 Trans. Electron Devices 59 (8) (2012) 2206-2214. DOI: 10.1109/TED.2012.2197951

[29] 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.N. 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 Trans. Electron Devices 62 (11) (2015) 3498-3507. DOI: 10.1109/TED.2015.2439635

[30] 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 521 (7550) (2015) 61-64. DOI: 10.1038/nature14441

[31] P. Yao, H. Wu, B. Gao, S.B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S.P. Wong, and H. Qian, Face classification using electronic synapses, Nat. Commun. 8 (2017), 15199. DOI: 10.1038/ncomms15199

[32] 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 Lett. 12 (5) (2012) 2179-2186. DOI: 10.1021/nl201040y

[33] 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: IEDM Tech. Dig. 2015, pp. 443-446 (2015). DOI: 10.1109/IEDM.2015.7409716

[34] 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, Front. Neurosci. 10 (2016), 56. DOI: 10.3389/fnins.2016.00056

[35] S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, and W. Lu, Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (4) (2010) 1297-1301. DOI: 10.1021/nl904092h

[36] 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 Trans. Electron Devices 58 (8) (2011) 2729-2737. DOI: 10.1109/TED.2011.2147791

[37] 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 22 (25) (2011), 254023. DOI: 10.1088/0957-4484/22/25/254023

[38] S. Ambrogio, S. Balatti, F. Nardi, S. Facchinetti, and D. Ielmini, Spike-timing dependent plasticity in a transistor-selected resistive switching memory, Nanotechnology 24 (2013), 384012. DOI: 10.1088/0957-4484/24/38/384012

[39] Z.-Q. Wang, S. Ambrogio, S. Balatti and D. Ielmini, A 2-transistor/1-resistor artificial synapse capable of communication and stochastic learning for neuromorphic systems, Front. Neurosci. 8 (2015), 438. DOI: 10.3389/fnins.2014.00438

[40] S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, N. Ramaswamy, and D. Ielmini, Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM, IEEE Trans. Electron Devices 63 (4) (2016) 1508-1515. DOI: 10.1109/TED.2016.2526647

[41] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, and E. Eleftheriou, Stochastic phase-change neurons, Nat. Nanotechnol. 11 (2016) 693-699. DOI: 10.1038/nnano.2016.70

[42] M.D. Pickett, G. Medeiros-Ribeiro, and R.S. Williams, A scalable neuristor built with Mott memristors, Nat. Mater. **12** (2013) 114-117. DOI: 10.1038/nmat3510

[43] X. Zhang, W. Wang, Q. Liu, X. Zhao, J. Wei, R. Cao, Z. Yao, X. Zhu, F. Zhang, H. Lv, S. Long, and M. Liu, An artificial neuron based on a threshold switching memristor, IEEE Electron Device Lett. 39 (2) (2018) 308-311. DOI: 10.1109/LED.2017.2782752

[44] G. Pedretti, V. Milo, S. Ambrogio, R. Carboni, S. Bianchi, A. Calderoni, N. Ramaswamy, A. S. Spinelli, and D. Ielmini, Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity, Sci. Rep. 7 (2017), 5288. DOI: 10.1038/s41598-017-05480-0

[45] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278-2324. DOI: 10.1109/5.726791

[46] A. Graves, A. Mohamed and G. Hinton, Speech recognition with deep recurrent neural networks, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing 2013, pp. 6645-6649. DOI: 10.1109/ICASSP.2013.6638947

[47] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 1097-1105 (2012).

[48] https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html

[49] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, Nature 521 (2015) 436-444. DOI:10.1038/nature14539

[50] S. Balatti, S. Larentis, D. Gilmer and D. Ielmini, Multiple memory states in resistive switching devices through controlled size and orientation of the conductive filament, Adv. Mater. 25 (10) (2013) 1474-1478. DOI: 10.1002/adma.201204097

[51] A. Prakash, J. Park, J. Song, J. Woo, E.-J. Cha, and H. Hwang, Demonstration of low power 3-bit multilevel cell characteristics in a TaO_x-based RRAM by stack engineering, IEEE Electron Device Lett. 36 (1) (2015) 32-34 DOI: 10.1109/LED.2014.2375200

[52] A. Athmanathan, M. Stanisavljevic, N. Papandreou, H. Pozidis, E. Eleftheriou, Multilevel-cell Phase-Change Memory: A viable technology, IEEE J. Emerging and Selected Topics in Circuits and Systems (JETCAS) 6 (1) (2016) 87-100. DOI: 10.1109/JETCAS.2016.2528598

[53] W.S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115-133.

[54] X. Glorot, A. Bordes and Y. Bengio, Deep sparse rectifier neural networks, in: Proc. 14th International Conference on Artificial Intelligence and Statistics 2011 vol. 15, pp. 315-323.

[55] S. Haykin, Neural networks and learning machines, 3rd Edition, Prentice Hall, 2009.

[56] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, H.-S. P. Wong, A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation, Adv. Mater. 25 (12) (2013) 1774-1779. DOI: 10.1002/adma.201203680

[57] J.-W. Jang, S. Park, G.W. Burr, H. Hwang, and Y.-H. Jeong, Optimization of conductance change in $Pr_{1-x}Ca_xMnO_3$-based synaptic devices for neuromorphic systems, IEEE Electron Device Lett. 36 (5) (2015) 457-459. DOI: 10.1109/LED.2015.2418342

[58] 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: IEDM Tech. Dig 2014, pp. 665-668. DOI: 10.1109/IEDM.2014.7047127

[59] S. Yu, P.-Y. Chen, Y. Cao, L. Xia, Y. Wang, and H. Wu, Scaling-up resistive synaptic arrays for neuro-inspired architecture: Challenges and prospect, in: IEDM Tech. Dig 2015, 451-454 (2015). DOI: 10.1109/IEDM.2015.7409718

[60] K. Moon, M. Kwak, J. Park, D. Lee, and H. Hwang, Improved conductance linearity and conductance ratio of 1T2R synapse device for neuromorphic systems, IEEE Electron Device Lett. 38 (8) (2017) 1023-1026 DOI: 10.1109/LED.2017.2721638

[61] S. Agarwal, R.B.J. Gedrim, A.H. Hsia, D.R. Hughart, E.J. Fuller, A.A. Talin, C.D. James, S.J. Plimpton, and M.J. Marinella, Achieving ideal accuracies in analog

neuromorphic computing using periodic carry, in: Symp. VLSI Tech. Dig 2017, pp. 174-175. DOI: 10.23919/VLSIT.2017.7998164

[62] F. Merrikh-Bayat, M. Prezioso, B. Chakrabarti, I. Kataeva, and D. Strukov, Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. arXiv preprint arXiv:1712.01253, 2017.

[63] S. Yu, Z. Li, P.-Y. Chen, H. Wu, B. Gao, D. Wang, W. Wu, and H. Qian, Binary neural network with 16 Mb RRAM macro chip for classification and online training, in: IEDM Tech. Dig 2016, pp. 416-419 (2016). DOI: 10.1109/IEDM.2016.7838429

[64] A. Coates, B. Huval, T. Wang, D. Wu, A.Y. Ng, and B.C. Catanzaro, Deep learning with COTS HPC systems, in: Proceedings of the 30th International Conference on Machine Learning 28 (3) (2013) 1337-1345.

[65] M. Hu, J.P. Strachan, Z. Li, R.S. Williams Dot-product engine as computing memory to accelerate machine learning algorithms, in: 17th International Symposium on Quality Electronic Design (ISQED) 2016, pp. 374-379. DOI: 10.1109/ISQED.2016.7479230

[66] 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: 53rd Annual Design Automation Conference (DAC) 2016, pp. 1-6. DOI: 10.1145/2897937.2898010

[67] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni and E. Eleftheriou, Mixed-precision in-memory computing, arXiv:1701.04279v4, 2017.

[68] Nandakumar S.R., M. Le Gallo, I. Boybat, B. Rajendran, A. Sebastian, and E. Eleftheriou, Mixed-precision training of deep neural networks using computational memory. arXiv:1712.01192v1, 2017.

[69] D. Saha, K. Leong, C. Li, S. Peterson, G. Siegel and B. Raman, A spatiotemporal coding mechanism for background-invariant odor recognition, Nat. Neurosci. 16 (2013) 1830-1839. DOI: 10.1038/nn.3570

[70] J. Humble, S. Denham, and T. Wennekers, Spatio-temporal pattern recognizers using spiking neurons and spike-timing-dependent plasticity, Front. Comput. Neurosci. 6 (2012), 84. DOI: 10.3389/fncom.2012.00084

[71] T. Deneux, A. Kempf, A. Daret, E. Ponsot, and B. Bathellier, Temporal asymmetries in auditory coding and perception reflect multi-layered nonlinearities, Nat. Commun. **7** (2016), 12682. DOI: 10.1038/ncomms12682

[72] G.-Q. Bi and M.-M. Poo, Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and post synaptic cell type, J. Neurosci. 18 (24) (1998) 10464-10472.

[73] G.M. Wittenberg, and S.S.-H. Wang, Malleability of spike-timing-dependent plasticity at the CA3-CA1 synapse, J. Neurosci. 26 (24) (2006) 6610–6617. DOI: 10.1523/JNEUROSCI.5388-05.2006

[74] L.F. Abbott and S.B. Nelson, Synaptic plasticity: Taming the beast. Nat. Neurosci. 3(Suppl) (2000) 1178–1183. DOI: 10.1038/81453

[75] S. Song, K.D. Miller and L.F. Abbott, Competitive Hebbian learning through spike-timing dependent synaptic plasticity, Nat. Neurosci. 3 (9) (2000) 919-926. DOI: 10.1038/78829

[76] N. Caporale and Y. Dan, Spike-timing dependent plasticity: A Hebbian learning rule, Annu. Rev. Neurosci. 31 (2008) 25-46 DOI: 10.1146/annurev.neuro.31.060407.125639

[77] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, and D. Ielmini, Statistical fluctuations in HfO_x resistive-switching memory: Part I – Set/Reset variability, IEEE Trans. Electron Devices 61 (8) (2014) 2912-2919. DOI: 10.1109/TED.2014.2330200

[78] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, and D. Ielmini, Statistical fluctuations in HfO_x resistive-switching memory: Part II – Random telegraph noise, IEEE Trans. Electron Devices 61 (8) (2014) 2920-2927. DOI: 10.1109/TED.2014.2330202

[79] S. Ambrogio, S. Balatti, V. McCaffrey, D. Wang, and D. Ielmini, Noise-induced resistance broadening in resistive switching memory – Part II: Array statistics, IEEE Trans. Electron Devices 62 (11) (2015) 3812-3819. DOI: 10.1109/TED.2015.2477135

[80] G. Pedretti, S. Bianchi, V. Milo, A. Calderoni, N. Ramaswamy, and D. Ielmini, Modeling-based design of brain-inspired spiking neural networks with RRAM learning synapses, in: IEDM Tech. Dig 2017, pp. 653-656. DOI: 10.1109/IEDM.2017.8268467

[81] G. Pedretti, V. Milo, S. Ambrogio, R. Carboni, S. Bianchi, A. Calderoni, N. Ramaswamy, A. S. Spinelli, D. Ielmini, Stochastic learning in neuromorphic hardware via spike timing dependent plasticity with RRAM synapses, IEEE J. Emerging Topics in Circuits and Systems (JETCAS) 8 (1) (2018) 77-85. DOI: 10.1109/JETCAS.2017.2773124

[82] R.M. French, Catastrophic forgetting in connectionist networks, Trends Cogn Sci. 3 (4) (1999) 128-135. DOI: https://doi.org/10.1016/S1364-6613(99)01294-2

[83] Z. Wang, S. Joshi, S.E. Savel'ev, H. Jiang, R. Midya, P. Lin, M. 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, Nat. Mater. 16 (2017) pp. 101-108. DOI: 10.1038/nmat4756

[84] V. Milo, D. Ielmini, and E. Chicca, Attractor networks and associative memories with STDP learning in RRAM synapses, in: IEDM Tech. Dig 2017, pp. 263-266. DOI: 10.1109/IEDM.2017.8268369