

Immunity to Device Variations in a Spiking Neural Network With Memristive Nanodevices

Damien Querlioz, *Member, IEEE*, Olivier Bichler, Philippe Dollfus, *Member, IEEE*, and Christian Gamrat

Abstract—Memristive nanodevices can feature a compact multilevel nonvolatile memory function, but are prone to device variability. We propose a novel neural network-based computing paradigm, which exploits their specific physics, and which has virtual immunity to their variability. Memristive devices are used as synapses in a spiking neural network performing unsupervised learning. They learn using a simplified and customized “spike timing dependent plasticity” rule. In the network, neurons’ threshold is adjusted following a homeostasis-type rule. We perform system level simulations with an experimentally verified model of the memristive devices’ behavior. They show, on the textbook case of character recognition, that performance can compare with traditional supervised networks of similar complexity. They also show that the system can retain functionality with extreme variations of various memristive devices’ parameters (a relative standard dispersion of more than 50% is tolerated on all device parameters), thanks to the robustness of the scheme, its unsupervised nature, and the capability of homeostasis. Additionally the network can adjust to stimuli presented with different coding schemes, is particularly robust to read disturb effects and does not require unrealistic control on the devices’ conductance. These results open the way for a novel design approach for ultraadaptive electronic systems.

Index Terms—Memristive devices, memristors, neuromorphic, spike timing dependent plasticity (STPD), spiking neural networks, unsupervised learning.

I. INTRODUCTION

MEMRISTIVE nanodevices provide fantastic opportunities for microelectronics. They can indeed provide a compact multilevel nonvolatile memory function [1]. However, they are often subject to strong variability [2]–[4], so that fully exploiting their potential would be easier with architectures offering a strong immunity to device variations. Spiking neural networks could provide a serious lead since the brain itself relies on variable neurons and synapses [5] and manages computational efficiency that outperforms man-made systems. This idea takes particular meaning in that many groups (constituting

the “neuromorphic” community) already imitate the brain with electronics, using Complementary Metal–Oxide Semiconductor (CMOS) circuits to model its spiking neurons and synapses. These studies, however, are often limited by the number of implementable synapses: implementing plastic synapses requires many transistors [6], [7]. Memristive nanodevices could provide the compact synapses required to advance neuromorphic circuits. In recent years several classes of them have indeed emerged, as e.g., resistive RAMs and memristors [1], [2], or adaptive transistors [3], [8]. It has been suggested [9]–[12], and shown experimentally [13]–[18], that such devices could reproduce a learning rule of biological synapses—spike timing dependent plasticity (STDP) [19], [20]—that is believed to be a foundation of learning in the brain [21]. A system consisting of nanoscale synapses and CMOS neurons could be a major breakthrough in computing, allowing cognitive-type tasks with high efficiency.

This idea is currently receiving considerable interest [22]–[25]. However, its sustainability is still to be demonstrated, especially with regards to the variability issue that is common to all memristive technologies [4], [26], [27]. It is also not clear if biological STDP is the best approach for electronics, the constraints of which differ strongly from Biology’s.

In this paper, system simulations introduce quantitative results in terms of computing performance and robustness to variability. We exploit a simplified and customized STDP scheme for memristive devices, which is a key to achieve effective learning with extreme robustness to memristive devices’ variability. It is associated with the use of unsupervised learning, and of a homeostasis-type mechanism. We describe the required technology and architecture (see Section II) and perform system-level simulations on a standard database of machine learning [28] that show the potential of the approach and its robustness. The system consists of an unsupervised layer that extracts features of the inputs using a simplified STPD (see Section III). The network performance compares favorably with traditional—but supervised—networks with similar numbers of adjustable parameters and achieves excellent tolerance to various memristive devices’ parameters variability. Finally, the robustness of the network to other device nonidealities (read disturb and limited resolution effects) is established (see Section IV).

Partial and preliminary results have appeared in [12]. This paper adds new discussions and results, especially the plausibility of the device model with regards to measurements of real devices, adaptations of the programming scheme, and the impact of diverse device nonidealities. Other proposals have been made to exploit variable adaptive devices in the context of nanotechnological implementations. Most proposed architectures

Manuscript received December 6, 2012; revised February 6, 2013; accepted February 19, 2013. Date of publication March 19, 2013; date of current version May 6, 2013. This work was supported in part by the European Union through the FP7 Project NABAB under Contract FP7-216777 and in part by the Centre National de la Recherche Scientifique/Institute for Engineering and Systems Sciences (PEPS Synapse). The review of this paper was arranged by Associate Editor D. Strukov.

D. Querlioz and P. Dollfus are with the Institut d’Electronique Fondamentale, University of Paris-Sud, Orsay 91405, France (e-mail: damien.querlioz@u-psud.fr; philippe.dollfus@ief.u-psud.fr).

O. Bichler and C. Gamrat are with the CEA, LIST, Laboratory for Enhancing Reliability of Embedded Systems, Gif-sur-Yvette 91191, France (e-mail: olivier.bichler@cea.fr; Christian.GAMRAT@cea.fr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNANO.2013.2250995

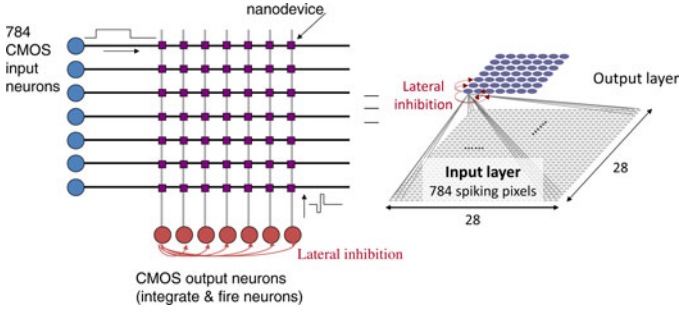


Fig. 1. Circuit topology. Wires originate from CMOS input layer (horizontal black wires) and from the CMOS output layer (vertical gray wires). Memristive nanodevices are at the intersection of the horizontal and vertical wires.

rely on reconfigurable logic [29] or on state-based supervised neural networks [30], [31]. In the first approach, variability is addressed through error mapping and redundancy, in the second case through a standard neural network approach using supervised learning based on the error gradient descent.

Our approach of using unsupervised learning with asynchronous spiking neural networks to tackle the variability issue of nanodevices is original and takes inspiration from recent ideas in computational neuroscience and neural networks [32]–[34]. Different works have been published that go into that direction. As mentioned previously, several proposals exist to use memristive devices for STDP [9]–[11], [35]. In this paper, we use a simplified STDP scheme that is easier to implement. Additionally, we propose to give to the neurons a homeostasis property and that is shown to be essential for the robustness of the scheme to variations. One work had already shown that memristive devices with STDP could allow the emergence of receptive fields in a variability-compatible unsupervised approach and synchronous neurons [36]. Our study uses asynchronous designs, like the ones used in the neuromorphic community [6], [7], and performs full learning on a standard dataset. Finally, an alternative way to allow learning with memristive devices in a variability-compatible way can be to use all digital designs [22]. This requires more devices per synapses [37]. In this paper, we show that variation-tolerance can be retained by using nanodevices with continuous variation of the conductance.

II. ARCHITECTURE AND IMPLEMENTATION OF THE NETWORK

We first introduce the architecture that we propose for our classifier system. The CMOS input and output “neurons” are connected by the nanodevices that act as synapses. It is natural to lay out the nanodevices in the widely studied crossbar as illustrated on Fig. 1, where CMOS silicon neurons and their associated synaptic driving circuitry are the dots, the squares being the nanodevices. The synapses indeed act as adaptive resistors. With the crossbar layout, if several synapses are active at the same time (i.e., receiving voltage spikes), the output receives directly the sum of the currents flowing through the synapses. In a more futuristic design, the system could also be laid out in a CMOL architecture where nanodevices crossbar is fabricated on top of the CMOS neurons and driving circuits [30].

This kind of connectivity corresponds to a feed-forward architecture in a neural network. Of particular interest, in the case of our network, it limits the sneak path issue because both programming and reading are performed in parallel. This issue usually limits the competitiveness of crossbars [38] and requires complex counter-measures like complementary resistive switches [39], or the use of nonlinear devices [40].

As a replacement of memristive devices, the architecture may also exploit phase change memories associated in “2-PCM” circuits as evidenced experimentally in [14], [18], and [41], which has the advantage of technological maturity.

The input neurons present the stimuli as asynchronous voltage spikes using several possible coding schemes described in Section III-C. The spiking rate is proportional to stimulus intensity. These stimuli may originate for example directly from a spiking retina [42] or cochlea [43] designed in the neuromorphic community that present data as asynchronous spikes, similarly to their biological counterparts.

As a result of learning, the output neurons should become selective to the different stimuli classes that are presented in a fully unsupervised manner: the output neurons should develop selectivity to specific features contained in the input patterns. The learning rule of the nanodevices needs to be fully local to be implementable in the crossbar architecture. The behavior of the neurons needs to be simple to make it easy to implement in a compact way. We now describe how this can be achieved.

The next section describe the different elements into more details. The synapses learn using a simplified STDP scheme (see Section II-A1). The output neurons behave as leaky integrate-and-fire neurons (see Section II-B1) and have a homeostasis property (see Section II-B3). They are also connected by inhibitory connections using diffuser networks (see Section II-B2).

A. Synaptic Behavior

1) *Simplified STDP Learning:* In this study, the synapses are acting in two ways. They are variable resistors and thus transmit spikes with a variable conductance (which plays the exact role of a synaptic weight). Additionally, they adapt their conductance depending on the activity of the neurons to which they are connected, which provides the foundation of learning by the system.

The memristive nanodevices are programmed as follows. When being applied a positive voltage pulse higher than a threshold V_{T+} , they increase their conductance. When applied a voltage pulse smaller than a negative threshold V_{T-} , they decrease their conductance [10], [13]. Previous works have shown that memristive devices can implement STDP, a learning rule used by brain synapses [9]–[11], [13]–[17], [35]. These works focused on faithful imitation of Biology. In this study, we focus on proposing simpler scheme, targeted toward pattern extraction, as illustrated in Fig. 2.

1) When an input neuron spikes, it applies a long voltage pulse to its synapses [see input pulse, Fig. 2(a)]. This voltage is high enough to drive some current into a memristive device, but not enough to reprogram it. This current is

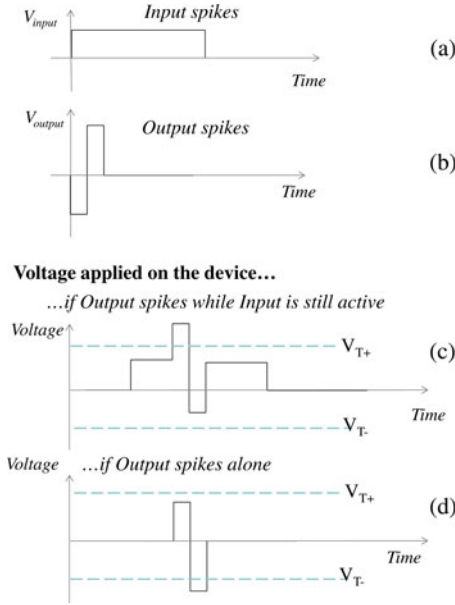


Fig. 2. Pulses for simplified STDP (voltage pulses as a function of time). When an input neuron spikes, it applies an input pulse (a) to the nanodevices to which it is connected. When an output neuron spikes it applies an output pulse (b) to the nanodevices to which it is connected. When the voltage applied on the device (difference between the voltages applied at the two ends (c) or (d)) reaches V_{T+} or V_{T-} , its conductance is increased or decreased, respectively.

integrated by the output neurons (resistor role of the synapse). If several synapses connected to the same output neuron are active at the same time their currents are summed.

- 2) When an output neuron spikes, it applies a pulse that is a succession of a negative bias and of a positive bias [see output pulse, Fig. 2(b)]. If no Input pulse is being applied to the device, only the second part reaches a threshold and the conductance of the synapse is decreased by δG_m [see Fig. 2(d)]. However, if the input neuron had spiked recently, and the input pulse is still being applied on the other end of the device, the voltage applied on the device actually increases its conductance by δG_p [see Fig. 2(c)].

This simple learning rule, easily implemented with nanodevices, is the ground for learning. Compared with the purely bioinspired and more complex scheme introduced in [9], no delay matching is necessary between the input and output synaptic waveforms, which should make the driving circuitry much easier to design.

The way this simple learning rule works is straightforward. When an output neuron declares a spike (at time t_{spike}), it increases by δG_p , the conductance of the synapses connected to input neurons that spiked recently (from $t_{\text{spike}} - t_{\text{PRE}}$ to t_{spike} , if t_{PRE} is the duration of the input pulse), and decreases by δG_m the conductance of the synapses that did not. This increases the sensitivity of the neuron to the specific pattern that activated it, making it more likely to spike for a similar (correlated) pattern in the future. This process—that works surprisingly well in practice, as we show in this paper—has been partially theo-

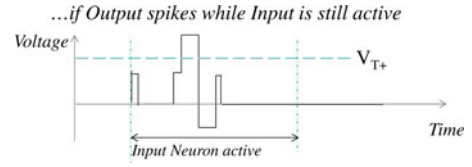


Fig. 3. Pulses for simplified STDP (voltage pulses as a function of time).

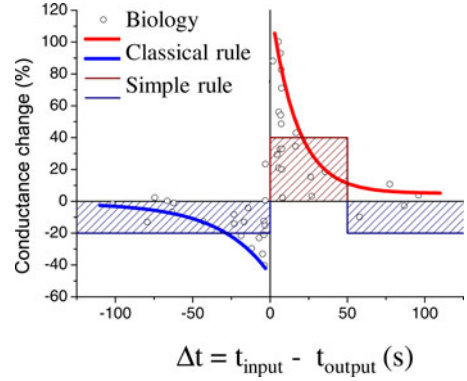


Fig. 4. Simplified STDP rule, compared with the standard biological STDP rule.

ried in [34]. A comparison with the traditional biological STDP scheme is presented in Fig. 4.

A disadvantage of this rule is the long input pulse, which drives current for a long time, and thus increases the power consumption. If it becomes significant, a low power version of the learning rule is possible and illustrated in Fig. 3, for the cost of limited added complexity. The input neuron does not apply a voltage pulse during the whole time it is active, but only at the beginning to drive current into the output neuron. Additionally, as soon as one of the output neuron becomes active, a signal is sent back to the input neurons, which apply a short input pulse again if they are still active. This lower power version of the learning scheme is more realistic with most current technologies like [13] and [14].

2) *Memristive Devices Modeling*: To model the conductance increments and decrements in our system simulations, we use the model introduced in [44]. It takes inspiration from experimental memristive devices measurements [13], [15]. An increase in the conductance is modeled by the equation

$$\delta G_p = \alpha_p e^{-\beta_p \frac{G - G_{\min}}{G_{\max} - G_{\min}}} \quad (1)$$

Similarly, a decrease is modeled by

$$\delta G_m = \alpha_m e^{-\beta_m \frac{G_{\max} - G}{G_{\max} - G_{\min}}} \quad (2)$$

The exponential factor expresses the fact, observed in most memristive technologies, that a given voltage pulse has a reduced effect on the device conductance if applied several times [13]–[15]. Agreement with the experimental data of [13] is presented in Fig. 5. The parameters α_p , α_m , β_p , and β_m depend heavily on the input and output pulse voltages that are chosen. These parameters, as well as minimum and maximum conductances G_{\min} and G_{\max} are subject to device variability in real devices.

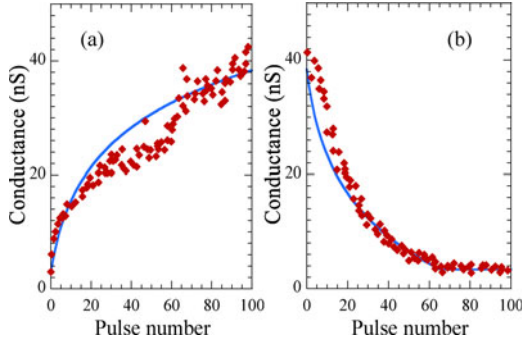


Fig. 5. Evolution of the conductance for the devices from [13], fitted with (1) and (2). (a) Device conductance (measured at 1 V) after each pulse in a series of potentiating ($V = -3.2$ V) pulses. (b) same with depressing ($V = 2.8$ V) pulses. Diamond: experimental data, reproduced from [13]. Full line: (1) and (2).

B. Output Neurons

1) *Output Neurons' Dynamics*: Exploiting the devices requires connecting them to processing units — silicon neurons able to process and generate spikes in a bioinspired manner by integration of their input. We call X the state variable (a current or a voltage, equivalent of the biological “membrane potential”) of the neuron (expressed in normalized unit where the maximum value of the state variable X is 1). Neurons are leaky integrate-and-fire type, which is meant to solve the simple following equation:

$$\tau \frac{dX}{dt} + gX = \gamma I_{\text{input}} \quad (3)$$

where τ is a leak time constant, and g and γ are constants. I_{input} represents the current flowing through the line of the crossbar connected to the neuron:

$$I_{\text{input}} = \sum_j I_j \quad (4)$$

where I_j are the currents flowing through each memristive device j connected to output neuron.

The neuron declares a spike if X reaches a given threshold X_{th} , in which case X is reset to zero.

An approach widely studied in the neuromorphic community is to use analog circuits (generally with transistors operating in the subthreshold regime) able to receive and generate asynchronous spikes [6], [7] to design such neurons. This kind of CMOS design is particularly low power, because most transistors operate in the subthreshold regime, and thanks to the use of asynchronous computation. Though smaller than in nanodevices, variability is also a problem for such transistor designs. It is a challenge for any neuromorphic design [43] and will become even more crucial when scaling to modern technology. Digital designs may also be used that do not suffer from variability directly but may have higher area and power requirements [45].

2) *Output Neurons' Lateral Inhibition*: When an output neuron spikes, it sends inhibitory signals to the other output neurons of the layer that prevent them from spiking during the inhibition time and resets their potential to zero. With this inhibition, the network is reminiscent of a winner-takes-all topology [34].

More precisely, when an output neuron spikes, the state variable X of the other output neurons is reset to zero during a time t_{inhibit} .

$$X = 0 \text{ if } t_{\text{spike}} < t < t_{\text{inhibit}}. \quad (5)$$

In hardware, this inhibition between the neurons can be implemented in a compact way through diffuser networks as in [15], which require a minimum number of transistors.

3) *Homeostasis*: A final issue for the architecture is the adjustment of the neurons' threshold. Simple algorithms exist for traditional (nonspiking) artificial neuron networks, but they do not work directly for spiking neurons. A bioinspired original route is homeostasis [5]. A target activity is defined for the neurons (i.e., a number of times an output neuron should spike over an extended period of time, like 100 digits presentation). Regularly, the threshold of the neuron is increased if the average activity of the neuron is above the target, and decreased if it is below

$$\frac{dX_{\text{th}}}{dt} = \gamma (A - T) \quad (6)$$

where A is the mean activity (or firing rate) of a neuron, T is the target activity, and γ is a multiplicative positive constant.

This ensures that all the output neurons are used and adjust the neurons' thresholds to the stimuli for which they become specialized. In neuromorphic hardware, homeostasis has been implemented with analog memories like in [46] or could be implemented digitally. The advantage inherent in this technology is evidenced in this paper in Section III-B.

C. Simulations of the System

In this paper, all simulations are system level and are based on a C++ special purpose code “Xnet” [12], [47]. The code is event-based for simulation performance and runs on traditional central processing units (CPUs). Simulation parameters as introduced previously are $\tau = 100$ ms, $g = 1$, $X_{\text{th}} = 0.5$ (normalized unit), inhibition time $t_{\text{inhibit}} = 10$ ms, $\alpha_p = 10^{-2}$, $\alpha_m = 5 \cdot 10^{-3}$, $G_{\text{min}} = 10^{-4}$, $G_{\text{max}} = 1$ (normalized units where the mean of the maximum conductance value is 1), $\beta_p = 3.0$ and $\beta_m = 3.0$. The width of the input pulses is 25 ms. Parameter variations are introduced around all the parameters using Gaussian random numbers (the value of their standard deviation is given in Section III). The initial conductances are selected randomly around mid range (0.5). The stimuli are applied using the coding schemes described in Section III-C.

The two variations of voltages pulses (see Figs. 2 and 3) gave identical performance in terms of recognition rate. All the results presented in the paper use the pulses of Figs. 2.

For demonstration of the concept, in this paper, we use the widely studied case of handwritten number recognition. The MNIST database is used, which consists in handwritten number of 28×28 pixels by 250 writers [28].

In order to achieve learning, we present the full MNIST training database (60 000 digits) three times to the system. Each input neuron is connected with one pixel of the image. It emits spikes with a jittered rate that is proportional to the pixel intensity (maximum rate is 22 Hz) as illustrated in Fig. 11(b). The initial

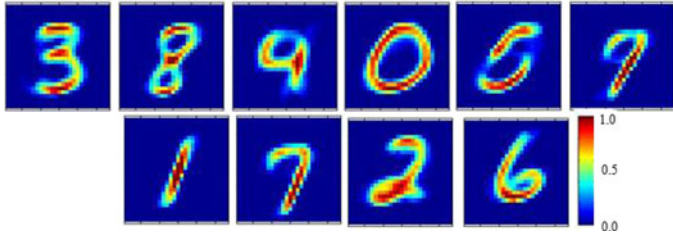


Fig. 6. Conductances (weights) learned in a simulation with ten output neurons. Red is maximum conductance and blue is minimum conductance.

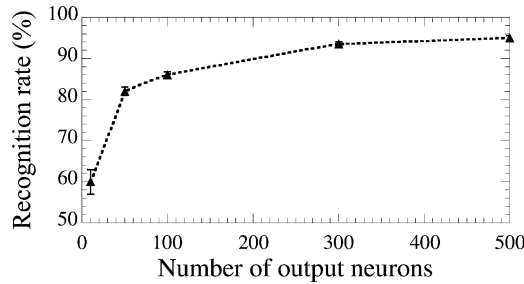


Fig. 7. Recognition rate on the test dataset as a function of the number of output neurons (simulations were repeated ten times, error bar is one standard deviation).

phase is random. Input neurons present spikes corresponding to a given digit during a period of 350 ms, after which they present spikes corresponding to another digit. No kind of preprocessing on the digits is used and the set is not augmented with distortions. The network is then tested on the MNIST test database, which consists of 10 000 digits that have not been presented during training. Simulation time was about 8 hours per run on an AMD Opteron 2216 CPU.

Fig. 6 plots the synaptic conductances (or weights) learned by the system in a configuration with only ten output neurons. It is remarkable that without any supervision and using only our local custom STDP rule, the system has identified nine (out of ten) different numbers, the real features of the input. Moreover, it has learned the distinctive features of the digits (and not just the most likely handwriting): it has learnt the loop of the digit two, the bottom of the six, or the horizontal parts of three and eight.

To evaluate the capability of the system, we can define a recognition rate. For that purpose, we associate output neurons with the digit for which they spike the most frequently *a posteriori*, using a subset of 1000 well-identified numbers. In hardware, this association could be performed with complex digital circuitry. An alternative can be to associate the unsupervised network with a supervised one [48]. All the simulations were repeated ten times, and the recognition rate given is averaged on the ten runs.

In order to evaluate quantitatively the network's recognition rate, Fig. 7 plots the final recognition rate on the test database. With ten output neurons the recognition rate reaches 60%. To improve the recognition rate, we can introduce additional output neurons, in which case some output neurons respond to different handwritings of the same digit. All the output neurons respond to some kind of digits, as explained in Section II-B3. With 50

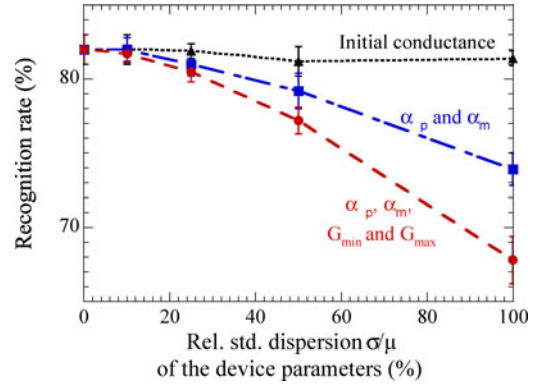


Fig. 8. Recognition rate on the MNIST testing set with different kinds of nanodevice (synaptic) variability. All the simulations were repeated ten times, the error bar is one standard deviation. All the simulations have homeostasis.

output neurons, the recognition rate reaches 81%, and with 300 output neurons 93.5%. A traditional artificial neural network with back propagation and 300 hidden neurons (obtained with the same number of adjustable parameters) reaches 95% [28], which compares to our rate of 93.5%. In the literature, the best algorithm has a largely superior 99.7% recognition rate, but using 12 million adjustable parameters (versus 235 200 here) and a largely augmented training set [49]. Though our numbers are clearly more modest, the interest here is that the network is fully unsupervised, with simple local learning rules, and variability immunity as is seen in the next section.

Thanks to the unsupervised nature of learning, the complicated “labeling” step does not need to be performed right away. Lots of unknown data may be presented for training, and the labeling can be performed from a limited subset of well-identified data. This is a strong advantage for many problems involving natural data. In many cases, a lot of data is available, but that was not classified, and the system can here perform this analysis by itself.

III. VARIABILITY IMMUNITY

A. Synaptic Variability

We now exploit our system-level simulations to analyze the robustness of our approach. We first study the impact of nanodevices (synaptic) variability on the recognition rate of the network. For this study, we simulated the network with 50 output neurons, introducing different kinds of variability. Results are reported in Fig. 8.

In the top curve of Fig. 8, we evaluate the impact of variability of the initial conductance (i.e., the initial conductance of the nanodevices, before learning starts). We can see it has no impact on the final recognition rate, even when the variability is extreme. This means that we do not need to control the initial conductance precisely. This is fortunate since this would be complex: controlling the exact state of memristive devices organized in a crossbar is a difficult issue, due to the device variations and the sneak paths.

In the middle curve of Fig. 8, we evaluate the impact of the variations on the learning increments and decrements [the

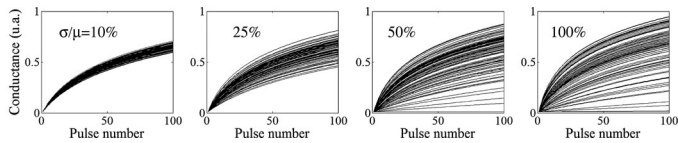


Fig. 9. Conductance as a function of pulse number (plotted similarly to Figure 5), for 100 devices with different relative standard dispersion σ/μ on the parameters α_p, α_m (from left to right 10, 25, 50, and 100%).

parameters α_p, α_m of (1) and (2)]. These variations can be caused by dispersion of the device thresholds and of their programming characteristics. We can see that the network is immune to variations up to 25% of the mean value of these parameters (i.e., $\sigma/\mu = 25\%$). This is already an extremely high level of variation for an electron device, but typical for research nanodevices. With 50% of variation, there is a small decrease of recognition rate (79% instead of 82% with no variability). With an extreme variability of 100% on the synaptic parameters, the recognition rate decreases significantly, but interestingly the functionality of the network is not challenged. We should note that with a variability of 50%, as many as 4% of the nanodevices cannot be programmed in at least one direction (i.e., they have an α value of 0). The latter figure becomes 30% when parameter variability is 100%. This is an indication of the overall tolerance to defects. To give a feel of the magnitude of these dispersions, we plotted on Fig. 9, the conductance as a function of pulse number (plotted similarly to Fig. 5), for 100 devices with different relative standard dispersion σ/μ on the parameters α_p and α_m .

The bottom curve of Fig. 8 adds variations of G_{\min} and G_{\max} . In that case some nanodevices overpower the others due to increased maximum conductance. Results are however similar to the previous case, and extreme robustness is preserved: with 50% variability, the recognition rate is lowered from 81.9% to a reasonable 77.2%. The most sensitive parameter is the maximum conductance. It should be noted that memristive currently devices developed in academic groups may have relative standard variations in G_{\min}, G_{\max} in the range between 50% or 100% [4], [26, suppl. info.]. The situation is expected to improve dramatically when the technology becomes mature. However, some significant degree of variability might be intrinsic to their physics [27].

This degree of robustness to device variation is exceptional in electronic systems and constitutes one of the strongest points of the approach. It takes root in the unsupervised nature of learning: output neurons learn features for which they are naturally suited. Synaptic variability determines how a neuron reacts initially, and thus what it is going to learn. This means that, in a way, variability is not a real problem, but a feature to initiate learning.

The simplicity of the voltages pulses is also fundamental to ensure this robustness.

B. CMOS Neuron Variability

We now study the impact of the variability of the CMOS neurons. The impact of the variability of their threshold is seen in Fig. 10. We can see that without homeostasis the impact

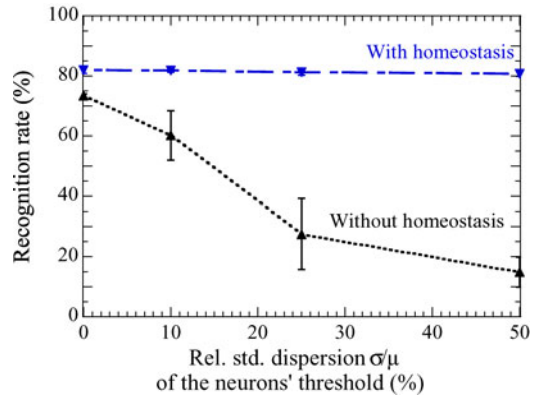


Fig. 10. Impact of the neurons' threshold X_{th} variability, with and without homeostasis on the recognition rate. All the simulations were repeated ten times.

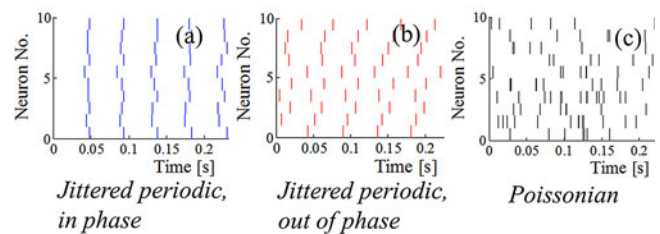


Fig. 11. Illustration of the three stimulus coding schemes. Stimulus spikes timing for ten input neurons with the same pixel value in the three schemes. All three lead to a recognition rate between 81% and 82%, without any parameter needing to be adjusted.

of the variability is dramatic. The more excitable neurons (the ones with lower thresholds) spike predominantly and the other neurons do not specialize efficiently. In a typical simulation with a threshold variation of 25%, the most excitable neuron spikes 51% of the time, and in a simulation with a threshold variation of 50%, it spikes 91% of the time. Homeostasis, however, fully compensates this issue (all the neurons spike between 1.5 and 3% of the time, most neurons spiking around 2%). And the same recognition rate as without threshold variation is achieved as evidenced in Fig. 10. Homeostasis appears as particularly valuable in this application. This also stresses the importance to evaluate the robustness of the system not only to nanodevices' variations, but also to the CMOS'.

C. Stimulus Encoding

Another remarkable point is the insensitivity to stimulus encoding scheme. Different schemes are possible to encode the stimulus into spikes as illustrated in Fig. 11. In the first two schemes [see Fig. 11(a) and (b)], the input neurons spike periodically, the firing rate being proportional to the input quantity. The input neurons can spike either in phase [see Fig. 11(a)] or out of phase [see Fig. 11(b)]. This is a natural scheme based on how spiking retinas currently work [42]. In the last case [see Fig. 11(c)], input neurons spike in a Poissonian way, the time constant being inversely proportional to the input. All three coding styles lead to a recognition rate between 81% and 82%. The only condition for this is that the window for potentiation in the STDP scheme is long enough with regards to the typical

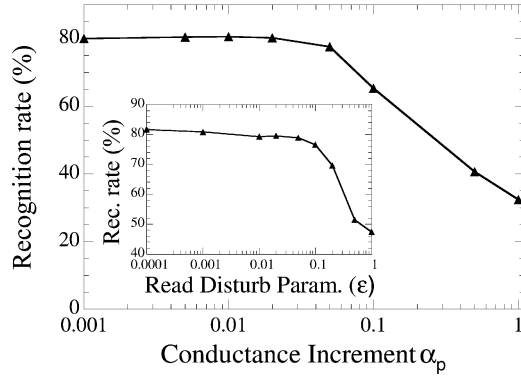


Fig. 12. Impact of the conductance increment on the recognition rate for a system with 50 outputs. Inset: Impact of the read disturb parameter ϵ on the recognition rate for a system with 50 outputs (as defined by equation (7)). Every simulation was repeated five times.

interspike interval. In this study, the average interspike interval is 45 ms for a black pixel, and insensitivity to stimulus encoding scheme is achieved if the window for potentiation is longer than 25 ms (while 15 ms is sufficient in the jittered periodic case). This shows again the robustness of the approach.

Additionally, noise can be inserted in the input with no impact. Adding 10% of perfectly random spikes distributed on all the input neurons has no impact on the system's recognition rate. This kind of robustness to spikes' precise timing is consistent with computational neuroscience works on STDP [32]. This result is not surprising since the precise timing of the spikes does not play a major role neither for the neurons nor for the simplified STDP learning rule.

IV. DEVICE NONIDEALITIES

Finally, we check that our approach is not sensitive to two crucial nonidealities of real memristive devices: the limited resolution of the conductance and the read disturb effect.

A. Limited Resolution of the Memristive Devices

The nanodevices that we are considering have an “analog memory” property. It has been evidenced in many real-life devices [13]–[17]. In these real devices, however, the conductance cannot be programmed with an arbitrary precision [13]. It is thus important to clarify which precision is required for our application. To that purpose, we performed simulations with different increments α_p of (1) (the decrement α_m of (2) is kept proportional to α_p). The results in terms of recognition rate are plotted in Fig. 12 (α_p is expressed in normalized units where the maximum conductance is 1). We see that recognition rate is maximum for α_p until 0.05, which is achievable by several experimental devices. For example the best measurements in [15] achieve $\alpha_p \sim 0.01$, or $\alpha_p \sim 0.02$ in [14], and $\alpha_p \sim 0.07$ in [13]. For higher values of α_p , the recognition rate decreases and the devices' resolution is not sufficient.

B. Impact of Read Disturb

Another significant device nonideality can materialize if the input pulses (which are subthreshold) have an impact on the device's conductance. This corresponds to the read disturb phenomenon for memory devices. To assess the impact of this phenomenon, we performed simulations where the input pulses increase the device conductance by a fraction ϵ of the increase of programming threshold

$$\delta G_{\text{subthreshold}} = \epsilon \delta G_p. \quad (7)$$

The results are presented in Fig. 12. It appears that read disturb parameter ϵ as high as 0.1 (meaning that a read pulse has 10% of the impact of a write pulse) may be tolerated, since it is fully compensated by learning. Such a value is a lot higher than what is seen in real devices [13], [26]. Read disturb should thus not be an issue for our approach.

V. CONCLUSION

In this study, using system-level simulations, we have shown how, by using a simple custom Spike Timing Dependent Plasticity scheme, memristive devices associated with CMOS neuromorphic circuits could perform unsupervised learning in a way that is extremely robust to variability. (A relative standard dispersion σ/μ of 50% is tolerated on all device parameters.) Nonsupervision is the foundation for this tolerance and provides extreme adaptability to the system. Homeostasis was also introduced as an essential component to retain insensitivity to CMOS variability. Additionally, the system is particularly tolerant to the read disturb effect, and does not require unrealistic control of the memristive devices' conductance.

This design approach could be the groundwork for future circuits that can process natural data in a compact and low power way. Thanks to their unsupervised learning capabilities, such circuits will be able to adapt to various environments. Future work should focus on the experimental demonstration of these concepts beyond single devices, and to demonstrate its scaling to more complex multilayer networks, and to other kinds of sensory stimuli like video, auditory, or olfactory data.

REFERENCES

- [1] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
- [2] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, “High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm,” *Nanotechnology*, vol. 23, p. 075201, Feb. 2012.
- [3] G. Agnus, W. Zhao, V. Derycke, A. Filoramo, Y. Lhuillier, S. Lenfant, D. Vuillaume, C. Gamrat, and J.-P. Bourgoin, “Two-terminal carbon nanotube programmable devices for adaptive architectures,” *Adv. Mater.*, vol. 22, no. 6, pp. 702–706, 2010.
- [4] 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 Lett.*, vol. 12, no. 1, pp. 389–395, 2011.
- [5] E. Marder and J.-M. Goaillard, “Variability, compensation, and homeostasis in neuron and network function,” *Nat. Rev. Neurosci.*, vol. 7, no. 7, pp. 563–574, 2006.
- [6] J. V. Arthur and K. A. Boahen, “Learning in silicon: Timing is everything,” *Adv. Neural Inf. Process. Syst.*, vol. 18, pp. 281–1185, 2006.

- [7] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 211–221, Jan. 2006.
- [8] F. Alibart, S. Pleutin, D. Guérin, C. Novembre, S. Lenfant, K. Lmimouni, C. Gamrat, and D. Vuillaume, "An organic nanoparticle transistor behaving as a biological spiking synapse," *Adv. Funct. Mater.*, vol. 20, no. 2, pp. 330–337, 2010.
- [9] J. Perez-Carrasco, C. Zamarreno-Ramos, T. Serrano-Gotarredona, and B. Linares-Barranco, "On neuromorphic spiking architectures for asynchronous STDP memristive systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 1659–1662.
- [10] G. Snider, "Spike-timing-dependent learning in memristive nanodevices," in *Prof. IEEE Int. Symp. Nanoscale Archit. (NANOARCH)*, 2008, pp. 85–92.
- [11] A. Afifi, A. Ayatollahi, and F. Raissi, "Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits," in *Proc. Eur. Conf. Circuit Theory Design*, 2009, pp. 563–566.
- [12] D. Querlioz, O. Bichler, and C. Gamrat, "Simulation of a memristor-based spiking neural network immune to device variations," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 1775–1781.
- [13] 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.*, vol. 10, no. 4, pp. 1297–1301, 2010.
- [14] M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction," in *Proc. IEEE Electron. Devices Meet. Tech. Dig.*, Dec. 2011, pp. 4.4.1–4.4.4.
- [15] 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, Jun. 2011.
- [16] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, and H. P. Wong, "An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation," *IEEE Trans. Electron Devices*, vol. 58, no. 8, pp. 2729–2737, Aug. 2011.
- [17] 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.*, vol. 12, no. 5, pp. 2179–2186, 2012.
- [18] 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*, vol. 59, no. 8, pp. 1–9, Aug. 2012.
- [19] H. Markram, J. Lubke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, pp. 2206–2214, Jan. 1997.
- [20] G.-Q. Bi and M.-M. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annu. Rev. Neurosci.*, vol. 24, pp. 139–166, Mar. 2001.
- [21] Y. Dan and M.-M. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron*, vol. 44, pp. 23–30, Sep. 2004.
- [22] G. Snider, R. Amerson, D. Carter, H. Abdalla, M. S. Qureshi, J. Léveillé, M. Versace, H. Ames, S. Patrick, B. Chandler, A. Gorchetnikov, and E. Mingolla, "From synapses to circuitry: Using memristive memory to explore the electronic brain," *Computer*, vol. 44, no. 2, pp. 21–28, 2011.
- [23] D. S. Modha, R. Ananthanarayanan, S. K. Esser, A. Ndirango, A. J. Sherbondy, and R. Singh, "Cognitive computing," *Commun. ACM*, vol. 54, pp. 62–71, Aug. 2011.
- [24] Y. V. Pershin and M. Di Ventra, "Neuromorphic, digital and quantum computation with memory circuit elements," *Proc. IEEE*, vol. 100, no. 6, pp. 2071–2080, Jun. 2012.
- [25] K. Cantley, A. Subramaniam, H. Stiegler, R. Chapman, and E. Vogel, "Hebbian learning in spiking neural networks with nanocrystalline silicon TFTs and memristive synapses," *IEEE Trans. Nanotechnol.*, vol. 10, no. 5, pp. 1066–1073, Sep. 2011.
- [26] 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*, vol. 464, no. 7290, pp. 873–876, 2010.
- [27] S. Yu, X. Guan, and H.-S. Wong, "On the stochastic nature of resistive switching in metal oxide RRAM: Physical modeling, monte carlo simulation, and experimental characterization," in *Proc. IEEE Int. Electron Devices Meet.*, Dec. 2011, pp. 17.3.1–17.3.4.
- [28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, Nov. 1998, vol. 86, no. 11, pp. 2278–2324.
- [29] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, "A defect-tolerant computer architecture: Opportunities for nanotechnology," *Science*, vol. 280, pp. 1716–1721, 1998.
- [30] J. H. Lee and K. K. Likharev, "Defect-tolerant nanoelectronic pattern classifiers," *Int. J. Circuit Theory Appl.*, vol. 35, pp. 239–264, May 2007.
- [31] D. Chabi, W. Zhao, D. Querlioz, and J.-O. Klein, "Robust neural logic block (NLB) based on memristor crossbar array," in *Proc. IEEE/ACM Int. Symp. Nanoscale Archit.*, 2011, pp. 137–143.
- [32] R. Guyonneau, R. VanRullen, and S. J. Thorpe, "Neurons tune to the earliest spikes through STDP," *Neural Comput.*, vol. 17, no. 4, pp. 859–879, 2005.
- [33] J. M. Brader, W. Senn, and S. Fusi, "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Comput.*, vol. 19, pp. 2881–2912, Nov. 2007.
- [34] B. Nessler, M. Pfeiffer, and W. Maass, "STDP enables spiking neurons to detect hidden causes of their inputs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1357–1365.
- [35] S. Yu, B. Gao, Z. Fang, H. Y. Yu, J. F. Kang, and H. S. Wong, "A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling," in *Proc. IEEE Int. Electron Devices Meet. Tech. Dig.*, 2012, pp. 10.4.1–10.4.4.
- [36] G. S. Snider, "Self-organized computation with unreliable, memristive nanodevices," *Nanotechnol.*, vol. 18, p. 365202, Sep. 2007.
- [37] G. Snider, "Instar and outstar learning with memristive nanodevices," *Nanotechnology*, vol. 22, p. 015201, Jan. 2011.
- [38] J. Liang and H. S. Wong, "Cross-point memory array without cell Selectors—Device characteristics and data storage pattern dependencies," *IEEE Trans. Electron Devices*, vol. 57, no. 10, pp. 2531–2538, Oct. 2010.
- [39] E. Linn, R. Rosezin, C. Kugeler, and R. Waser, "Complementary resistive switches for passive nanocrossbar memories," *Nat. Mater.*, vol. 9, no. 5, pp. 403–406, 2010.
- [40] J. Joshua Yang, M. X. Zhang, M. D. Pickett, F. Miao, J. Paul Strachan, W.-D. Li, W. Yi, D. A. A. Ohlberg, B. Joon Choi, W. Wu, J. H. Nickel, G. Medeiros-Ribeiro, and R. S. Williams, "Engineering nonlinearity into memristors for passive crossbar applications," *Appl. Phys. Lett.*, vol. 100, p. 113501-1–113501-4, Mar. 2012.
- [41] M. Suri, O. Bichler, D. Querlioz, B. Traoré, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Physical aspects of low power synapses based on phase change memory devices," *J. Appl. Phys.*, vol. 112, pp. 054904-1–054904-10, Sep. 2012.
- [42] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x 128 120 dB 15 MUS latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.
- [43] V. Chan, S.-C. Liu, and A. van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Trans. Circuits Syst., Regul. Pap.*, vol. 54, no. 1, pp. 48–59, Jan. 2007.
- [44] D. Querlioz, P. Dollfus, O. Bichler, and C. Gamrat, "Learning with memristive devices: How should we model their behavior?" in *Proc. IEEE/ACM Int. Symp. Nanoscale Archit.*, 2011, p. 150.
- [45] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. Modha, "A digital neurosynaptic core using embedded crossbar memory with 45 pJ per spike in 45 nm," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2011, pp. 1–4.
- [46] C. Bartolozzi, O. Nikolayeva, and G. Indiveri, "Implementing homeostatic plasticity in VLSI networks of spiking neurons," in *Proc. IEEE Int. Conf. Electron., Circuits Syst.*, 2008, pp. 682–685.
- [47] O. Bichler, D. Querlioz, S. J. Thorpe, J.-P. Bourgoin, and C. Gamrat, "Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity," *Neural Netw.*, vol. 32, pp. 339–348, 2012.
- [48] D. Querlioz, W. S. Zhao, P. Dollfus, J.-O. Klein, O. Bichler, and C. Gamrat, "Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches," in *Proc. IEEE Int. Symp. Nanoscale Archit.*, 2012, pp. 203–212.
- [49] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep big simple neural nets excel on handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, Mar. 2010.

Authors, photographs and biographies not available at the time of publication.