Modeling and Implementation of Firing-Rate Neuromorphic-Network Classifiers with Bilayer Pt/Al2O3/TiO2-x/Pt Memristors

M. Prezioso1, I. Kataeva2,§, F. Merrikh-Bayat1, B. Hoskins1, G. Adam1, T. Sota2, K. Likharev3‡, and D. Strukov1†

1 UC Santa Barbara, Santa Barbara, CA 93106-9560, U.S.A.
2 Research Laboratories, DENSO CORP., 500-1 Minamiyama, Komenoki-cho, Nisshin, Japan 470-0111
3 Stony Brook University, Stony Brook, NY 11794-3800, U.S.A.

email: *mprezioso@ece.ucsb.edu, §irina_kataeva@denso.co.jp, †strukov@ece.ucsb.edu, ‡konstantin.likharev@stonybrook.edu

Abstract

Neuromorphic pattern classifiers were implemented, for the first time, using transistor-free integrated crossbar circuits with bilayer metal-oxide memristors. 10×6- and 10×8-crosspoint neuromorphic networks were trained in-situ using a Manhattan-Rule algorithm to separate a set of 3×3 binary images: into 3 classes using the batch-mode training, and into 4 classes using the stochastic-mode training, respectively. Simulation of much larger, multilayer neural network classifiers based on such technology has sown that their fidelity may be on a par with the state-of-the-art results for software-implemented networks.

Introduction

Deep-learning convolutional neural networks, which are essentially multilayer perceptrons (Fig. 1a) with restricted connectivity between some layers, have been demonstrated to achieve some of the best classification performances on a variety of benchmark tasks (1). The major challenge in building fast and energy-efficient networks of this type in hardware is performing efficient vector-by-matrix multiplication, which in turn requires compact implementation of synaptic weights (2).

CrossNet circuits have emerged as an efficient solution to these challenges (3). In such a network, neural cell bodies (somas) are mimicked with analog CMOS circuits, which communicate via passive crossbars with integrated tunable resistive devices ("memristors") (4-7), playing the role of synapses (8-12) – see Figs. 1b-e. Two main goals of this work were to demonstrate the first neural networks with integrated crossbar circuits, and evaluate possible performance of larger classifiers based on this emerging technology. (Some preliminary results were reported in (10).)

Memristive crossbar circuit

A 12×12 crossbar with 200-nm lines separated by 400-nm gaps (Fig. 2a), with a Ta/Pl/Al2O3/TiO2-x/Ti/Pt memristor at each crosspoint, was fabricated using a standard lift-off patterning. The Al2O3/TiO2-x stack was deposited by reactive sputtering, with titanium oxide stoichiometry controlled precisely via the oxygen flow control. The thickness and stoichiometry were optimized to achieve low forming voltages (<2 V) and highly nonlinear I-V curves with a ~10 ratio of current values at the switching voltage (~1.5 V) and at a half of it (Fig. 2b). The most outstanding feature of such memristors is their low variability (Fig 2c); together with nonlinear I-V and low forming voltages it has enabled forming of most of the devices in crossbar array. Other important characteristics are the ~100 ON/OFF current ratio at ~0.1 V, a switching endurance of at least 5,000 cycles, an estimated retention of at least 10 years at room temperature, and operation currents between ~100 nA and ~100 μA (10).

Using short (e.g., 500 μS) pulses makes both set and reset switching processes fairly continuous, enabling gradual tuning of device conductance with an at least 5-bit precision even using a very simple (suboptimal) feedback algorithm (11) – see Fig. 3. Such precision is already acceptable for some neural network applications (3, 12).

Experimental results

During classifier's operation (Figs. 1e, 4, 5a), the vector-by-matrix multiplication of the input signals (represented with voltages) by weights (represented by memristor conductances) is performed on the physical level, in analog domain, using Ohm’s and Kirchhoff’s laws, by applying the input voltages to crossbar’s row lines and reading out the currents flowing into virtually grounded column lines (Figs. 1e, 5a). The training was performed in-situ in both the batch and stochastic modes, using the Manhattan-Rule algorithm (13) - see Fig. 4. This rule is convenient for crossbar circuit implementation, due to the use of only the sign information of the conventional Delta-Rule algorithm's result.

The advantage of stochastic training is that the weight update for the whole crossbar (of any size) may be performed in just four steps by applying pulses in parallel to rows and columns of the crossbar (12) – see Fig. 5b. Namely, the weights are grouped into four sets, each corresponding to a particular combinations of signs of V(n) and δ(n), and the weight in each group are updated in parallel. On the contrary, in the batch mode the weights in different columns (or rows) have to be updated sequentially (Fig. 5c), so that the update time grows linearly with crossbar size. Additionally, the batch...
mode training may come with a large area overhead when implemented on-chip, due to the need of computing and storing intermediate results for the weight update (14).

Generally, device-to-device variations of the switching threshold present a significant challenge for the in-situ training, because exponential switching dynamics (7, 11) amplifies even slight threshold variations. Additionally, the change in conductance depends on the initial conductance of the device. In this context, the fact that we have been able to achieve successful convergence for both the batch and stochastic in-situ training (Fig. 6), even despite substantial device-to-device variations in switching dynamics (Fig. 7), is highly encouraging. The batch-mode training gave more stable convergence, and the update dynamics for stochastic training was very close to that in the software-implemented network (Fig. 8).

Simulation results
In another part of this work, an accurate, data-verified model of adaptation of our memristors (15) was used to simulate the performance of pattern classifiers, based on a large-scale fully connected multilayer perceptron and a deep-learning convolutional network, on several representative benchmarks (16), using both in-situ and ex-situ training. Similarly to the experimental results, the classification performance was worse for the stochastic Manhattan-Rule training (Table 1a). However, a simple “variable-amplitude” variation (Fig. 9) of the training scheme (Fig. 5b-c) allows an implementation of the more efficient Delta-Rule algorithm (Fig. 4b), which dramatically improves the stochastic-mode fidelity (Table 1) and achieves state-of-the-art performance for the batch training. In such variable-amplitude scheme, write voltages proportional to $\log[V(n)]$ and $\log[\delta(n)]$, of specific polarity, are applied to the corresponding lines of the crossbar. Since the change of device conductance is roughly exponential in the applied voltage, this procedure results in weight update proportional to the product of $\delta V$, thus implementing the Delta Rule directly in the crossbar, without the need of its calculation in external hardware. The simulation results also show that the in-situ training is inherently robust to various network defects (Fig. 10), and that an 8-bit weight import at ex-situ training is sufficient to avoid classification fidelity degradation (Fig. 11).

Conclusions
In summary, we have experimentally demonstrated an artificial neural network using memristors integrated into a dense, transistor-free crossbar circuit. We believe that this demonstration is a significant step toward analog-hardware implementation of practically useful artificial neural networks. The simulation of such scaled-up networks, using a quantitatively verified model of our memristors, has shown that their performance can be competitive to the state-of-the-art software implementations. Moreover, recent experiments (17) with similar but smaller (so far, discrete) memristors give hope that the metal-oxide memristor networks may be scaled down to at least 30-nm devices. According to theoretical estimates (3), such networks would enable CrossNets with an areal density higher than that of the human cerebral cortex, operating at much higher speed and with comparable energy efficiency.

Acknowledgements
This work was supported by AFOSR under MURI grant FA9550-12-1-0038, by DARPA under contract HR0011-13-C-0051 UPSIDE via BAE Systems, Inc., and by the DENSO CORP., Japan.

References

(9) S. Park et al., “RRAM-based synapse for neuromorphic system with pattern recognition function”, IEDM Technical Digest, p. 10.2.1, 2012.
Fig. 1. Neuromorphic network implementation with CrossNet circuits [3]: (a) A graph representation of a multilayer perceptron; (b) a cartoon of a hybrid CMOS/memristor (CMOL) integrated circuit; (c) analog implementation of the dot-product, (f) its mapping on the hybrid circuit, and (e) the implementation of vector-by-matrix multiplication using a memristive crossbar. (It shows that if negative weight values are required, a synapse may be implemented as a pair of memristors.)

Fig. 2. Crossbar circuit with integrated Al₂O₃/TiO₂ resistive switching devices: (a) micrograph of a 12×12 crosspoint crossbar; (b) typical quasi-de I-V curves of memristor forming and switching, with the inset showing the device stack; and histograms of: (c) conductances before forming, (d) forming voltages, and (e) effective switching threshold voltages. (The threshold is conditionally defined as the point at which device’s resistance is changed by at least 2 kΩ upon application of a 500-μs voltage pulse train with a slowly increasing amplitude, starting from high/low conductive state for reset/set transitions.)

Fig. 3. Analog properties of crossbar-integrated devices: (a) tuning of the resistance measured at a non-disturbing voltage of 0.1 V (sampled at 1 kHz and averaged over 200 s interval) to various values within the dynamic range, (b) a repeated state measurement over time, and (c) an example of tuning sequence. Error bars on panel (a) show the standard deviations during the time sequence shown on panel (b). The sudden changes of resistance, visible on panel (b), are more noticeable at higher resistances.

Fig. 4. In-situ training of a single-layer perceptron classifier: (a) flow chart of one epoch for batch- and stochastic-mode training algorithms. Gray-shaded boxes show the steps implemented inside the crossbar, while those with solid black borders denote the only steps required for performing the feedforward (classification) operation.
Fig. 5. Physical-level description of the classification experiment: (a) example of operation of classifier using a 10 × 6 fragment of the crossbar; example of weight adjustment for (b) stochastic and (c) batch training for a specific error matrix. Panels (b) and (c) show the voltages only for first two steps. The read and write biases were always $V_D = 0.1$ V and $V_{I_{bias}} = \pm 1.3$ V, respectively (Fig. 2b).

![Image](image_url)

Fig. 6. Results of pattern classification experiments: the convergence of network’s output in the process of in-situ training for the (a) batch and (b) stochastic training modes; (c)-(d): the training and test images used for (c) batch and (d) stochastic training experiments. For the batch training, one epoch is the input of 30 patterns, while for stochastic training, one iteration is the application of one pattern. The batch mode training (test) images are formed by flipping one pixel (two pixels) of the “ideal patterns” shown with the solid border.

![Image](image_url)

Fig. 7. Average change in resistance, measured at 0.1V, for 18 devices inside the crossbar when applying one (a) reset and (b) set pulse. Error bars show the standard deviation.

![Image](image_url)

Fig. 8. Update dynamics table for the 10×8 crossbar, showing the average number of times a device was set (counted as +1) and reset (-1) during the stochastic training: (a) simulation results for software-based perceptron averaged over 500 runs, and (b) experimental results, for similar device initialization.

![Image](image_url)

Fig. 9. An example of a “variable-amplitude” four-step weight update for 2×2 crossbar and specific $V_i(n)>0$, $V_D(n)=0$, $\delta_i(n)>0$, and $\delta_D(n)=0$.

![Image](image_url)

Fig. 10. MNIST dataset classification fidelity of a multilayer perceptron as a function of the fraction of stuck-on-open or stuck-on-close devices, for several training approaches.

![Image](image_url)

Fig. 11. Classification performance - as a function of weight import precision (simulated by adding normally distributed noise) for a deep-learning convolutional network [16] trained by an ex-situ training method.

![Image](image_url)

Table 1. Classification fidelity for (a) 300-hidden-neuron multilayer perceptron network tested on the MNIST benchmark, and (b) a deep-learning convolutional neural network tested on three indicated benchmarks. The deep network architecture is similar to those described in [14]. 500 patterns per batch were used for batch training.