Comparison of two memristor based neural network learning schemes for crossbar architecture

Janusz A. Starzyk[1, 2] and Basawaraj[1]

[1] School of Electrical Engineering and Computer Science, Ohio University, Athens, OH, USA
[2] University of Information Technology and Management, Rzeszow, Poland
{starzykj, basawaraj.basawaraj.1}@ohio.edu

Abstract. This paper compares two neural network learning schemes in crossbar architecture, using memristive elements. Novel memristive crossbar architecture with dense synaptic connections suitable for online training was developed. Training algorithms and simulations of the two proposed learning schemes, winner adjustment training (WAT) and multiple adjustments training (MAT) are presented. Tests performed using MNIST handwritten character recognition benchmark dataset confirmed the functionality of proposed learning schemes. Proposed learning schemes were compared accounting for noise, device variations and multipath effects. The proposed learning schemes improve available neural network learning schemes.

Keywords: neural network, memristive crossbar, self-organizing, learning, dense analog memories, character recognition

1 Introduction

Memristors, first hypothesized by Chua 40 years ago [1], are elements relating charge and magnetic flux and change their resistance based upon the input current or voltage. Resistance switching in Titanium dioxide (TiO_2) was first recognized as related to the memristor characteristics in 2008 [2] and sparked considerable interest in their potential applications, including neural networks [3 – 6]. Later work has shown the existence of memristor characteristics in WO_3 [7] and other metal oxides like ZrO_2, NiO, Nb_2O_5, HfO_2, and CeO_x [8].

The ability to perform on-line training and scale the architecture to a large system are the major evaluation criterion of memristive architectures in neural network applications. The goal of this work is to make memristive on-line learning feasible in high density crossbar architectures. Specifically, we pro-

pose a neural network organization and two training scheme, suitable for high density architectures with interconnection weights that can be easily controlled during the network operation.

Neural networks using high-density crossbar architecture have been proposed in earlier work [9 – 11]. The architecture proposed in [9] is easily scalable as it is based on a simple crossbar structure with memristors formed at the junctions. But [9] lacks in details regarding mechanism for training the memristors in the crossbar architecture, probably due to its emphasis on fabrication of memristors. [10] proposes a neural crossbar architecture, provides a training scheme and shows its application to design of fault tolerant circuits using supervised learning. A related design in [11], shows application of a crossbar architecture using memristors for handwritten character recognition. Unlike [11] our approach uses neither lateral inhibition among the output neurons nor homoeostasis at the output neurons to dynamically adjust their thresholds. This decreases the circuitry needed and results in a more compact architecture.

2 Memristor Characteristics

Memristors change their resistance as a function of the flux applied to their terminals and remember their "state" when the flux is removed. Fig. 1 shows a physical model of the memristor as described in [2]. It consists of titanium dioxide (TiO_2) and an oxygen poor TiO_{2-x} layer, fabricated between platinum contacts. The titanium dioxide layer is considered the undoped region and has high resistance while the oxygen poor region is considered the doped region and has low resistance due to the oxygen vacancies acting as positive dopants. The effective resistance is the sum of the resistances of the doped and undoped regions:

$$M(x) = R_{ON}x + R_{OFF}(1 - x) \tag{1}$$

where the state variable

$$x = \frac{\mathrm{w}}{D} \in (0,1) \tag{2}$$

And R_{ON} and R_{OFF} are the resistance of the doped region and undoped regions respectively. The memristor's current and voltage are related through Ohm's law, with memristance M(x) representing current resistance value.

Fig. 1. (a) Memristor model according to [2] (b) memristor symbols used in this paper left-to-right: traditional symbol, symbols used in this paper for p-type and n-type memristors.

Memristor is an asymmetrical device and its properties depend on the direction of the current that passes between its terminals. Conductivity increases when a positive voltage is applied across the memristor (plus at doped region and minus at undoped region), otherwise its conductivity goes down. Thus both the voltage value and duration of the applied voltage are important, so the resistance is in effect a function of total flux as well as initial state of the memristor.

Experiments and analysis performed following publication of [2] has furthered the understanding of the memristor characteristics. For example, [12] predicted that the speed of switching ON was faster than switching OFF; applying ON-switching for sufficiently long time would turn the device OFF i.e., the switching polarity can be reversed. These devices have ohmic i-v characteristics near the ON states and are non-linear towards the OFF states. Moreover, in [13] it was shown that increasing the applied current caused an exponential decrease in the required switching energy. The model from [2], used here, though limited is still a very good approximation of memristor characteristics and was used due to its simplicity and wide use in literature. Using a more detailed model, would not considerably affect either the algorithm or the results presented here.

3 Design of Neural Networks

To be compatible with neural network learning schemes the memristor conductance values are normalized to [0-1] interval and are used to represent

weights of synaptic connections. For simplicity only excitatory neurons are considered here. A simple two layer neural network is shown on Fig. 2 (a). In this network neuron N is a postsynaptic neuron and it receives inputs from n presynaptic neurons x_1,… x_n. Here, for convenience, we assume that the output of an excitatory neuron is V_{out} = 1 V when activated and V_{out} = 0 V otherwise.

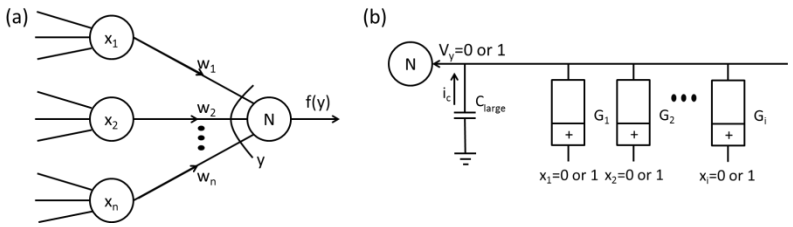

Fig. 2. (a) Two layer feed forward neural network; (b) Memristors connected to a neuron's input.

Fig. 2 (b) shows an input to a neuron N in a crossbar architecture with a number of memristor elements connected to its input line. When the neuron N's input in Fig. 2 (b) is charged to the voltage V_y above a threshold value V_{th}, the neuron fires and sets its feedback output voltage to V_{yf} for a short period of firing time T_f, regulated by adjusting load capacitance C_{large}. Thus, with 1 volt on the active input V_i the corresponding memristor increases its conductance (weight) whereas the inactive input (V_i =0 V) reduces its conductance. Here V_y can be obtained from

$$V_y = \sum \frac{V_i G_i}{\sum G_i} = \sum w_i V_i \tag{3}$$

where

$$w_i = \frac{G_i}{\sum G_i} \tag{4}$$

3.1 Neural Network Training

According to Hebbian learning, in a simple feedforward neural network, excitatory links increase their weights when both presynaptic and postsynaptic neurons fire. We adopt this rule to the memristor circuit with different

schemas of weight adjustment and resulting training algorithms. Neural network training in crossbar architecture algorithm is as follows.

Neural network training in crossbar architecture.

1. Select s random training data and use them to assign initial weights to the output neurons.
2. For all training samples repeat 3 - 6.
3. Compute similarity between training data and neuron weights using (3) and obtain input excitations of the output neurons V_y.
4. Sort all output neurons excitations V_y to determine dynamic threshold value V_{th} for the neurons' input excitation V_y.
5. Provide training feedback value V_{yf} = 1 V to all output neurons y with $V_y < V_{th}$ and provide training feedback value V_{yf} = 0 V to all output neurons y with $V_y > V_{th.}$ That is, if the output neuron, N, fires, it uses feedback to reduce its input potential V_{yf} to zero. Thus with 1 volt on the active input V_i, the corresponding memristor increases its conductance (weight). However, if N does not fire, then it changes its input potential V_{yf} to 1 V for a period of firing time T_f. This produces a negative flux across memristor connected to the inactive input with V_i = 0 V reducing its conductance (weight). Neuron's firing time is regulated by adjusting its load capacitance C_{large}.
6. Adjust memristor weights using $\Delta w_i = (V_i - V_{yf})\Delta_i(G)$. $\Delta_i(G)$ is the increment of Δw_i per unit voltage applied across the memristor. V_{yf} is the provided feedback signal $V_{yf} = \overline{V}_{out}$. The resulting weight adjustments are a result of the memristor voltage polarization. For instance if the input voltage $V_i = 0\ V$ and $V_{yf} = 1\ V$ memristor weight is lowered and when $V_i = 1\ V$ and $V_{yf} = 0\ V$, it increases.

Two versions of the neural network training algorithm are used. They differ by the way a threshold value is established in step 4 of the algorithm. The first version known as **multiple adjustments training** (MAT) uses median value of V_y to establish the threshold and at every step of the training cycle up to 50% of memristor values are adjusted upwards and up to 50% downwards. This balances off total amount of flux that each memristor receives. According to Step 6 the adjustment upwards takes place only when V_y = 0 V

and $V_i > 0$ V and the adjustment downwards takes place only when $V_y = 1$ V and $V_i < 1$ V.

MAT requires additional circuits to adjust threshold V_{th} dynamically. Its advantage is that training of memristors can be performed in dense crossbar architecture since no additional switches are required to select which memristors need to be adjusted upwards or downwards. Instead, memristors' control is done by adjusting threshold and feedback voltages only.

The second version known as **winner adjustment training** (WAT) selects a single winner that is most similar to the training sample. Thus V_{th} is equal to the winning neuron's V_y. Subsequently, all memristor connections are open except for the winning neuron, and the feedback voltage V_{yf} is set to 0.5 V. Thus, memristors of the winning neuron connected to inputs higher than 0.5 V are adjusted upwards and those lower that 0.5 V are adjusted downwards, moving neuron in the direction of the input data.

For compactness of the neural network design it is desirable that no elements other than memristors at the crossbar junctions are added to the crossbar architecture. This is the main advantage of the proposed approach. WAT requires only a winner-take-all circuit, thus it is easier to implement in hardware than MAT. However, since all memristors of losing neurons must be disconnected during training, WAT crossbar architecture must contain switching transistors in series with memristors to implement synaptic connections. All memristors in a single column (one output neuron) can be controlled by the same control signal, simplifying wiring of the control signals.

4 Benchmark Testing

In benchmark testing neural network has its memristive weights adjusted according to either WAT or MAT algorithm. We investigated test results for WAT and MAT training using MNIST [14] handwritten character database. Neural network with 784 inputs and 80 outputs was constructed and trained using 1280 data points. Once training was completed and memristor values were adjusted, we applied a new set of 10,000 test data from the MNIST database. The average test performance over 25 runs was 73.2% with stand-

ard deviation of 1.8% for WAT and 69.9% with standard deviation of 1.6% for MAT.

4.1 Analysis for Robustness

To test the robustness of the learning schemes we performed analysis to determine the effect of multipath, input noise, and memristor manufacturing tolerances. Memristors are not connected directly to ideal voltage sources but use drivers with nonzero resistance hence neural network training and testing will be subject to multipath effects. To estimate this we simplify our analysis to the one in which the input signals correspond to the input digit intensity and the feedback signal in the neural network training is obtained with V_y connected to a voltage source (0 V or 1 V) through a fixed conductance G_s. In such circuit, deviation of V_y from its ideal value (0 V or 1 V) will depend on the input signal as follows:

$$V_y = \frac{\sum_{k=1}^{m_c} V_i G_i + G_s V_s}{\sum_{k=1}^{m_c} G_i + G_s} \tag{5}$$

where m_c is the number of memristors in this category (either winners or losers). In case $V_s = 1$ we use (3) for memristors of losing neurons, and in this case deviation of the feedback voltage ΔV_y form a desired value equals

$$\Delta V_y = 1 - V_y = \frac{\sum_{k=1}^{m_l} G_i(1-V_i)}{\sum_{k=1}^{m_l} G_i + G_s} \tag{6}$$

where m_l is the number of memristors that are in the losing neurons. For $V_s = 0$ we use (3) for memristors of winning neurons, and in this case ΔV_y equals

$$\Delta V_y = V_y = \frac{\sum_{k=1}^{m_w} V_i G_i}{\sum_{k=1}^{m_w} G_i + G_s} \tag{7}$$

where m_w is the number of memristors that are in the winning neurons. Due to similar deviations on the driver side memristor training will be subject to a significant noise. In the worst case this noise is uncorrelated, so we tested how such noise will affect testing performance. It was observed that, for driver conductance range of 10 µS to 100 mS, the performance in both tested methods (MAT and WAT training) did not suffer.

To analyze the robustness of learning schemes the test data was corrupted with uniform noise signal. The analysis was performed using a single layer neural network with 784 inputs and with 250 output neurons. Fig. 3 shows the test results and shows that the recognition rate is better than chance even for noise to signal ratios of 10 (for this dataset the chance level is 10%).

The last test of robustness performed used Monte-Carlo analysis to test the ability of the proposed neural network learning scheme to handle process variations leading to changes in nominal values of all memristors. While memristors fabricated on different dies will have significant deviations from their nominal values, crossbar memristors are fabricated on the same die and will have memristance parameters tracking each other.

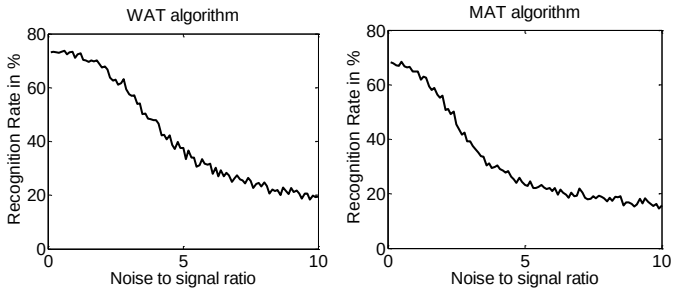

Fig. 3. Correct recognition performance on 10000 test data with various noise levels.

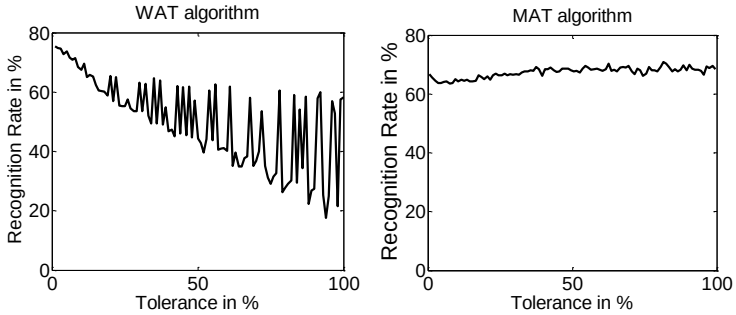

Fig. 4. Correct recognition performance in Monte Carlo analysis.

Absolute changes in memristive conductivity are not important as long as the ratio in (4) remains unchanged. Critical for neural network performance are local variations within the same die which are small and correlated, thus their effect on the performance of a neural network will be also small.

We performed 100 tests, each starting from the same training data set (based on 2000 randomly selected digits) and the same 10,000 test data points but with memristor characteristics randomly varied within specified tolerance. The neural network had 300 output neurons. The tolerances on memristor parameters were increased from 0.5% until 100% of their nominal values. Fig. 4 shows the test results. Results show that recognition performance for WAT quickly goes down, so the network is more sensitive to variations in memristor parameters that it was to the input noise. However, since typically, within the same die, the memristances do not vary more than 5% (a reasonable assumption for medium size dies with present manufacturing capabilities [15]), the loss of the recognition performance is small. There is no observable loss in performance in MAT training even at larger tolerances.

5 Conclusion

In this paper, we presented two memristor training schemes in the crossbar organization such that neural network interconnection weights can be easily implemented and controlled during the network operation. We performed simulations to test the proposed neural network learning schemes considering multipath effects, device variations and noisy test data. The results showed that the proposed approach is compact, tolerant to noise and device variations, and is can be trained online. The results obtained verified correct adjustment of memristor values for the selected training data. In this work, the focus was on developing neural network learning schemes for crossbar architecture, rather than on specific data bases or advanced neural network application. The proposed solution improves available in the literature training methods for memristive neural networks.

Acknowledgements. This work was supported by the grant from the National Science Centre DEC-2011/03/B/ST7/02518.

6 References

1. L. Chua, “Memristor-the missing circuit element,” IEEE Trans. Circuit Theory, vol. CT-18, no. 5, pp. 507–519, Sep. 1971.

2. D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” Nature, vol. 453, pp. 80–83, May 2008.

3. Y. V. Pershin and M. Di Ventra, “Experimental demonstration of associative memory with memristive neural networks,” Neural Netw., vol. 23, no. 7, pp. 881 – 886, Sept. 2010.

4. Y. V. Pershin, and M. Di Ventra, “Solving mazes with memristors: A massively parallel approach,” Phys. Rev. E, vol. 84, no. 4, pp. 046703, Oct. 2011.

5. K. D. Cantley, A. Subramaniam, H. Stiegler, R. Chapman, and E. Vogel, “Neural learning circuits utilizing nano-crystalline silicon transistors and memristors,” IEEE Trans. Neural Networks Learning Syst., vol. 23, no. 4, pp. 565–573, April 2012.

6. H. Kim, M. P. Sah, C. Yang, T. Roska, and L. O. Chua, “Memristor Bridge Synapses,” Proceedings of the IEEE, vol. 100, no. 6, pp. 2061 - 2070,June 2012.

7. T. Chang, S. -H. Jo, K. -H. Kim, P. Sheridan, S. Gaba, and W. Lu, “Synaptic behaviors and modeling of a metal oxide memristive device,” Appl. Phys. A, vol. 102, no. 4, pp. 857-863, Mar.2011.

8. L. Liu, B. Chen, B. Gao, F. Zhang, Y. Chen, X. Liu, Y. Wang, R. Han, J. Kang “Engineering oxide resistive switching materials for memristive device application”, Appl Phys A, no. 102: pp. 991–996, Feb. 2011.

9. 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. 1297–301, Apr. 2010.

10. D. Chabi, and J. -O. Klein, "Hight fault tolerance in neural crossbar," Design and Technology of Integrated Systems in Nanoscale Era (DTIS), 2010 5th International Conference on , vol., no., pp.1,6, 23-25 March 2010.

11. 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," Nanoscale Architectures (NANOARCH), 2012 IEEE/ACM International Symposium on , vol., no., pp.203,210, 4-6 July 2012.

12. D.B. Strukov, J.L. Borghetti, and R.S. Williams, “Coupled ionic and electronic transport model of thin-film semiconductor memristive behavior,” Small, vol. 5, no. 9, pp. 1058 – 1063, 2009.

13. M. Pickett, D.B. Strukov, J. Borghetti, J. Yang, G. Snider, D. Stewart, and R.S. Williams, “Switching dynamics in a titanium dioxide memristive device,” J. of App. Phys., vol. 106, art. 074508, 2009.

14. The MNIST database of handwritten digits. Available: http://yann.lecun.com/exdb/mnist/

15. D. T. Gray, “Optimization of the Process for Semiconductor Device Fabrication in the MicrON 636 Whittemore Cleanroom Facility,” M.S. thesis, Materials Sc. Eng., Virginia Polytechnic Inst. and State Uni., Blacksburg, VA, 2002.