Memristor-based neuron circuit and method for applying learning algorithm in SPICE

C. Yakopcic, R. Hasan, T.M. Taha, M. McLean and
D. Palmer

The learning of nonlinearly separable functions in cascaded memristor crossbar circuits is described and the feasibility of using them to develop low-power neuromorphic processors is demonstrated. This is the first study evaluating the training of memristor crossbars through SPICE simulations. It is important to capture the alternate current paths and wire resistance inherent in these circuits. The simulations show that neural network learning algorithms are able to train in the presence of alternate current paths and wire resistances. The fact that the approach reduces the area by three times and power by two orders of magnitude compared with the existing approaches that use virtual ground opamps to eliminate alternate current paths is demonstrated.

Introduction: Memristors [1] have received significant attention as a potential building block for neuromorphic systems [2, 3]. Physical memristors [4] can be laid out in a high density grid known as a crossbar [5]. Using this layout, memristors have the potential to be fabricated with a synaptic density greater than that of brain tissue [6]. Using these devices will produce high density, extreme low-power, neuromorphic hardware that is capable of performing many multiply–add operations in parallel in the analogue domain.

To the best of our knowledge, this Letter presents the first study of a learning algorithm being applied to a memristor-based crossbar neural circuit in SPICE (so alternate current paths are considered). Existing studies [7] have shown that using the virtual ground mode operational amplifiers at each column in the memristor crossbars eliminates alternate current paths. Therefore, higher level simulation tools can be used to analyse and train the neural networks. However, the opamps add significant area and energy requirements to the circuit compared with the circuit approach presented in this Letter. This makes the virtual ground approach undesirable for lower power applications.

In this Letter, we show that a dense memristor crossbar can be trained with neural network algorithms despite the sneak paths inherent in the circuits. A two-layer neural network is trained to demonstrate how nonlinearly separable functions can also be learned through our low-power circuits, proving that these neuromorphic circuits can learn any logic function. In this case, the multilayer circuit was trained using the concurrent learning algorithm (CLA) [8], which is a supervised learning algorithm that is designed to have a minimal hardware footprint.

Neural circuit: The schematic in Fig. 1*a* shows our approach for developing a memristor-based neuron circuit. This example shows three data inputs in addition to a high and low bias input. Each signal input is connected to the positive and negative comparator input through a grid of memristors. The neuron outputs in our study are thresholded, and thus a comparator is used. Two memristors are required to represent a single synaptic weight as this allows for each data input to have either a positive or negative effect on the total dot product (DP_j). Fig. 2 shows how the individual neurons in Fig. 1*a* will fit together to develop the multilayer network described in Fig. 1*b*. The memristors in this circuit are arranged in a crossbar to provide a very high synaptic density.

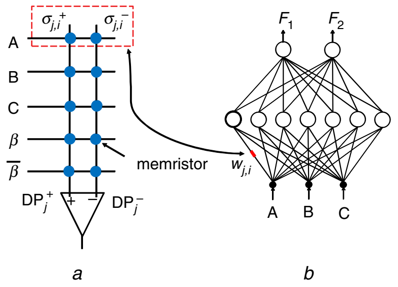

Fig. 1 *Circuit diagram for single memristor-based neuron and how it fits into neural network constructed in this Letter*
a Circuit diagram
b Neural network

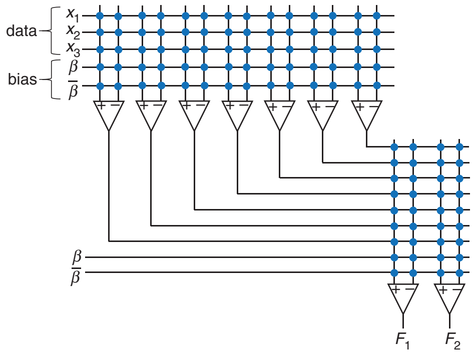

Fig. 2 *Schematic for neural network circuit that was simulated*

Training approach: Algorithm: The CLA [8] was utilised in training the circuit in this Letter. The steps in this algorithm to train the crossbar for a set of inputs x are:

1. Initialise the memristors with high random resistances.
2. Apply an input pattern x to the crossbar circuit in SPICE and evaluate hidden neuron and output neuron values. Record comparator inputs DP^+_j and DP^-_j.
3. Calculate the difference, del, between the comparator outputs (F_j) and the target outputs (D_j) using (1) in MATLAB

$$\mathrm{del} = \sum_j D_j - F_j \tag{1}$$

4. Determine the amount, Δw, that each memristor's conductance should be changed (η is the learning rate) using (2) and (3)

$$\mathrm{DP}_j = \mathrm{DP}^+_j - \mathrm{DP}^-_j \tag{2}$$

$$\Delta w_{j,i} = \eta \times \mathrm{del} \times \frac{1}{1 + \mathrm{DP}^2_j} \times x_{j,i} \tag{3}$$

5. Apply write pulses to the crossbar in SPICE with pulse widths proportional to $\Delta w_{j,i}$ to update each memristor conductance.
(6) Go to step 2 and apply the next input pattern x in the sequence.

Using this method for updating the weights replaces two steps in the training algorithm with analogue operations in the crossbar circuits (simulated in SPICE). First the dot products (DP_j) would normally be calculated according to (4) and (5). Furthermore, the weight update (6) and (7) are replaced by a series of pulses that alter the weights proportional to $\Delta w_{j,i}$

$$\mathrm{DP}^+_j = \sum_i x_i \left(w^+_{j,i} \right) \tag{4}$$

$$\mathrm{DP}^-_j = \sum_i x_i \left(w^-_{j,i} \right) \tag{5}$$

$$w^+_{j,i,\,\mathrm{new}} = w^+_{j,i,\,\mathrm{old}} + \Delta w_{j,i} \tag{6}$$

$$w^-_{j,i,\,\mathrm{new}} = w^-_{j,i,\,\mathrm{old}} - \Delta w_{j,i} \tag{7}$$

It is assumed that this entire training process will eventually be performed in hardware surrounding the crossbar circuit. The actual hardware design that will record the comparator inputs (or dot products) will be saved for future work.

To model the memristor device in SPICE, a previously published memristor model [9] was utilised that is capable of reproducing switching characteristics very accurately. The memristor device simulated for this circuit was published in [10] and the switching characteristics for the model are displayed in Fig. 3. This device was chosen for its high minimum resistance value, large resistance ratio and fast switching time.

As opposed to carrying out the simulation in MATLAB, evaluating the actual memristor circuit in SPICE allows for more accurate modelling of the memristor grid. The alternate current paths and wire resistance within the crossbar are simulated. Our results show that training a crossbar without the presence of alternate current paths leads to improper operation when the final weights are simply written to the memristors in the circuit in Fig. 2. Therefore, this design can be used for *in situ* training, but not *ex situ* training. Virtual ground mode operational amplifiers [7] can be used to eliminate alternate current paths (allowing for *ex situ* training), although our results show that using these opamp circuits leads to a 3× increase in circuit area and an 88× increase in energy consumption. Fig. 4 displays the total energy consumption per neuron of both the proposed approach and the virtual ground alternative.

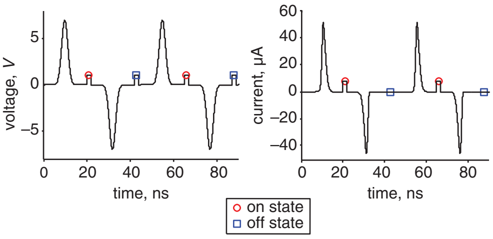

Fig. 3 *Simulation results displaying input voltage and current waveforms for memristor model [9] based on device in [10]*

These parameter values were used in the model to obtain this result: $Vp = 4$ V, $Vn = 4$ V, $Ap = 816\,000$, $An = 816\,000$, $xp = 0.985$, $xn = 0.985$, $\alpha p = 0.1$, $\alpha n = 0.1$, $a1 = 1.6 \times 10^{-4}$, $a2 = 1.6 \times 10^{-4}$, $b = 0.05$, $x0 = 0.01$

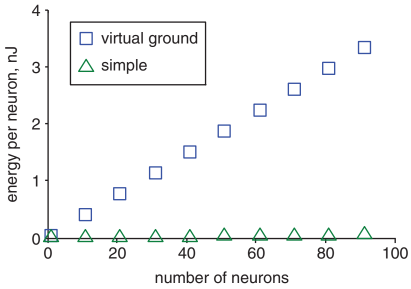

Fig. 4 *Energy consumption as number of neurons is increased*

Neural network simulation: The two output neurons were trained to recognise the nonlinearly separable functions displayed in (8) and (9). There is no plane in a three-dimensional space which can separate the different outputs of these functions. Therefore, at least a two-layer neural network with nonlinear thresholding is required

$$F_1 = A \oplus B \oplus C \tag{8}$$

$$F_2 = ABC + \bar{A}\bar{B}\bar{C} \tag{9}$$

Fig. 5 shows the absolute error present throughout the training process when learning functions F_1 and F_2. Learning continues until the total error reaches 0, and the results show that it took about 130 epochs to learn both functions.

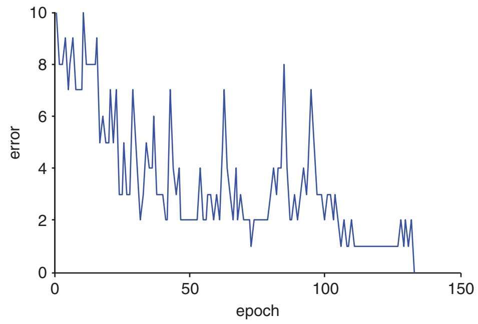

Fig. 5 *Result that displays training error reaching zero after about 130 epochs*

Conclusion: This Letter clearly shows that a passive memristive crossbar can be used as the synaptic component in larger neuromorphic processors. Simulations in SPICE have shown that the CLA algorithm [7] could successfully train a multilayer neuromorphic circuit in the presence of alternate current paths. This shows that a neuromorphic processing tile based on our design can learn any digital logic function.

11 February 2014
doi: 10.1049/el.2014.0464
One or more of the Figures in this Letter are available in colour online.

C. Yakopcic, R. Hasan and T.M. Taha (*University of Dayton, 300 College Park, Dayton, OH 45409, USA*)

E-mail: cyakopcic1@udayton.edu

M. McLean (*Center for Exceptional Computing, 5200 Research Park Drive, Catonsville, MD 21228, USA*)

D. Palmer (*Annapolis Microsystems,190 Admiral Cochrane Drive, Suite 130, Annapolis, MD 21401, USA*)

References

1 Chua, L.O.: 'Memristor – the missing circuit element', *IEEE Trans. Circuit Theory*, 1971, **18**, (5), pp. 507–519
2 Chabi, D., Zhao, W., Querlioz, D., and Klein, J.-O.: 'Robust neural logic block (NLB) based on memristor crossbar array'. IEEE/ACM Int. Symp. Nanoscale Architectures, San Diego, CA, USA, June 2011, pp. 137–143
3 Zamarreño-Ramos, C., Camuñas-Mesa, L.A., Pérez-Carrasco, J.A., Masquelier, T., Serrano-Gotarredona, T., and Linares-Barranco, B.: 'On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex', *Front. Neurosci. Neuromorphic Eng.*, 2011, **5**, pp. 1–22, Article 26
4 Strukov, D.B., Snider, G.S., Stewart, D.R., and Williams, R.S.: 'The missing memristor found', *Nature*, 2008, **453**, pp. 80–83
5 Jo, S.H., Kim, K.-H., and Lu, W.: 'High-density crossbar arrays based on a Si memristive system', *Nano Lett.*, 2009, **9**, (2), pp. 870–874
6 Snider, G.S.: 'Cortical computing with memristive nanodevices', SciDAC Review, 2008
7 Alibart, F., Zamanidoost, E., and Strukov, D.B.: 'Pattern classification by memristive crossbar circuits with ex-situ and in-situ training', *Nat. Commun.*, 2013, **4**, p. 2072
8 McLean, M.R.: 'Concurrent learning algorithm and the importance map', *Netw. Sci. Cybersecur. Adv. Inf. Secur.*, 2014, **55**, pp. 239–250
9 Yakopcic, C., Taha, T.M., Subramanyam, G., Pino, R.E., and Rogers, S.: 'A memristor device model', *IEEE Electron Device Lett.*, 2011, **32**, (10), pp. 1436–1438
10 Lu, W., Kim, K.-H., Chang, T., and Gaba, S.: 'Two-terminal resistive switches (memristors) for memory and logic applications'. Proc. 16th Asia and South Pacific Design Automation Conf., Yokohama, Japan, January 2011, pp. 217–223