Logic Computing in Memristor-based neural networks

Kai Bu∗, Haijun Liu, Wei Wang

College of Electronic Science and Technology, University of Defence Technology, 109 Deya Road, Changsha, Hunan, China

*Corresponding author Email: booquai@nudt.edu.cn

Abstract. This paper presents a neural network based on a memristor crossbar to realize linearly separable logic functions with a single-layer perceptron in just one step, while realizing linearly nonseparable functions with multilayer neural network in multiple steps, such as addition operation and XOR logic. We could implement different logic functions in the same crossbar and circuit by adjusting the resistance of the memristor. It’s conveniently changed by analog tuning of the applied voltages. As the neural networks are capable of tolerating the random diversity effects of device variations and noises to some degree, the proposed method can tolerate the variations of crossbar arrays. We also built a verification platform with Xilinx FPGA, Analog-Digital converters, and a 32x32 memristor crossbar array to show how the logic function works.

1. Introduction

The development of electronic information technology has entered the post-Moore era, the reducing the feature size of semiconductor devices is facing the dual constraints of the fabricating technics and the physical mechanism. And the performance improvement is also more and more difficult for processors designed following Von-Neumann computing architecture[1,2]. The "memory wall" is becoming a severe problem caused by the data access bottleneck between processor and memory. The CPU processor could not meet the requirements of performance and energy consumption for big data processing in the intelligent era[3,4,5]. The new Computing-in-Memory(CiM) architecture based on memristor technology is a meaningful way to break through the performance bottleneck of the traditional Von-Neumann computing system. Memristor has the potential feature to build a universal digital nonvolatile computing system to break the energy-efficiency bottleneck caused by the separation of storage and processing in the traditional computing framework. High-performance devices, efficient logic gates and reconfigurable architecture are the basis for improving nonvolatile computing systems[6,7,8]. As a new passive nanoscale information device, memristor has brought new possibilities to the design of computing architecture with its characteristics of both logic and storage. The CiM technology has built a new type of computing architecture, which breaks the separation between computing and storage, and is expected to solve the problem of memory-wall. Profit from its unique physical mechanism of resistance, memristor has become a valuable research highlight of integrated logic calculation based on the ability of both computing and storage[9,10].

2. Logic on Memristor crossbar

The resistance of the memristors can be switched between two digital states, low resistance state corresponding to logic-high ‘1’ and high resistance state corresponding to logic-low ‘0’, by two
different types and amplitude of voltage pulses[6]. Crossbar is usually employed as memory architecture based on memristor for high integration density. Further, a memristor with crossbar architecture, showing the outstanding characteristics of storing data and potential for parallel computing, is suitable for in-memory computing.

Recently, many efforts have been paid on the implementation of memristive logic gates for in-memory computing. Two categories, Voltage-Resistance (V-R) logic gate and Resistance-Resistance (R-R) logic gate, can be broadly divided by input and output physical variables. In the V-R logic gate, the input and output signals are voltage and resistance values, respectively. Additional register and latch circuits are required for the resistance-voltage converting process when the former logic gate output (R value) is employed as the post logic input (V value). Hence, this conversion process results in degrading of computation speed, increasing power consumption and circuit complexity of the computing system[11,12].

In the R-R logic gate (also called stateful logic), input and output signals are both represented by resistance state. Resistance interaction is the basic working principle of a stateful logic gate. Specifically, voltage excitation of the logic gate determines the type of logic operation. Memristor resistance (output) changes to a specific logic state (LRS or HRS) according to the voltage drop across it, which is affected by the input resistance state and the way of voltage excitation. Stateful logic schemes have been springing up by adding more memristors in parallel or changing the excitation signals’ amplitude, which is more competitive in complexity and operation delay.

![Figure 1. OR logic operations in R-R mode](image)

We take OR logic in R-R logic gate mode as an example. As shown in Figure 1, the voltage dropped on device R can be expressed as:

\[
V_R = \frac{VR}{R + \frac{RA}{RA + RB}}
\]

R, RA, and RB are the resistance of memristors A, B, and R, respectively. The voltage V is the excitation voltage of the unit. When the NAND operation is executed, the device R is initialized to logic '1'. For the ideal devices, if A and B are both in the HRS, VR can be expressed as:

\[
V_R = \frac{VR}{2} + \frac{RA}{2}
\]

Because the resistance of A is much higher than that of R. Then, VR is close to zero, and the resistance of R remains at LRS. When A and B are both in the LRS, VR is equal to 2V/3, which is larger than the threshold VRESET driving the device R to logic '0'. If one of A and B is in HRS and the other one is in LRS (here, A is assumed in LRS), VR can be expressed as:

\[
V_R = \frac{V}{2 + \frac{RB}{RA + RB}}
\]

RA/(RA + RB)is close to 1, because the resistance of A is much higher than that of R. Then, V_R is approximating V/2 which cannot change the low resistance state of the device R. The analysis of the resistance interaction in other operation is much the same.

Due to the same physical formations of input and output, sequentially cascading multiple operations are completed without converting the circuit out of memory, which can be ascribed true in-
memory computing process. However, unstable devices and some inefficient logic gates are the real drawbacks. Stable resistance state, abrupt and concentrated threshold voltage are requisites for the system based on resistance interaction, while excellent retention and endurance are necessary for the memory and computing function. In terms of the stateful logic scheme, the number of the logic gates with different functions is insufficient and the only set process is utilized to execute logic operations, which limits the flexibility of the system reconfiguration and date schedule. When a complex computing task is assigned, a massive number of devices and operation steps make the system inefficient. Also, up till now, we can not get an accurate circuit model and operation steps for a new computing function. We need to design them step by step racking our brains because there has not been a universal framework and method developed to implement different logic functions in the same circuit architecture.

3. Logic computing in Memristor-based neural networks

In recent years, the synapse of a neural morphological system based on memristor has been widely concerned. A memristor crossbar array can easily perform matrix multiplication, which is a very expensive operation for a neural network. Figure 2 shows the basic structure of the memristor crossbar array [13-16]. In a memristor crossbar array, there is a memristor at each crosspoint of a row and column. Voltage pulse could be transfer to one end of each memristor through its row, and a summation of output current could be generated along a column, while the electric conductance of each memristor is its weights based on the law of Kirchhoff’s [17-19].

![Figure 2. neural network mapped on memristor crossbar array](image)

The Boolean logic function is equivalent in function to a pattern classification operating, which could be processed by a perceptron. This paper presents a neural network based on a memristor crossbar to realize linearly separable logic functions with a single-layer perceptron in just one step, while realizing linearly nonseparable functions with multilayer neural network in multiple steps, such as addition operation and XOR logic. We could implement different logic functions in the same crossbar and circuit by adjusting the resistance of the memristor. It’s easily changed by tuning of the input voltage pulse. The circuit can be extended to a multilayer neural network to implement complex nonseparable operations in multiple steps that provide a universal framework and method to build a completed complete logic computing circuit based on memristor crossbar array[11].

![Figure 3. AND logic single-layer perceptron mapped on memristor crossbar array](image)
The logic computing functions based on a neural network could share the same basic structure, which mainly consists of a crossbar array with several resistive switch nodes. The resistance of the switch node could be viewed as input data of a neural network and the voltages applied on the node could be considered to be a weight of the input. When we want to carry out a different logic function, we just need to adjust the weights in the way of tuning the applied voltages can in the same circuit. The resistance of the input data remains unchanged, which means computing in-suit or computing in memory.

The variation of device-to-device and the yield of manufacturing are a big problem in the mass production of the memristor. Memristor’s performance may be degraded due to the test cycle and programming. These cases will significantly reduce the accuracy of resistance interaction calculation. The variation of memristor devices mainly includes switch changes and parameter changes. The driving circuit will also cause the random variation indifferent period of writing or reading, and the small fluctuates of programming voltage will lead to the significant change of memory resistance. Figure 4 shows the impact on the distributions of resistance states caused by imperfect manufacturing, oxide thickness variations, and random dopants.

![Figure 4. Variations impact on resistance distribution of memristor](image)

However, the neural networks can tolerant the variations of crossbar arrays to a certain degree as its weights are more robust. At the same time, we trained the neural networks with a data set with variation simulating to real memristor devices. The network for different logic has different tolerance abilities for the same variations, which caused different accumulative calculation errors. We inject a random error into the training data sets with a normal distribution according to real variations. By giving more training data sets of the values of memristor crossbar with real variations, we could get a better weight by significantly increase the learning accuracy. A software-based and hardware-based calibration method could be designed to improve the performance of the memristor-based neural network.

| Work      | XOR Devices | XOR Steps | Tolerance of Variations |
|-----------|-------------|-----------|-------------------------|
| Reference 13 | 4           | 5         | No                      |
| Reference 14 | 4           | 4         | No                      |
| Reference 17 | 3           | 2         | No                      |
| This paper  | 3           | 2         | Yes                     |

This universal framework and method could take the minimum and a fixed number of devices and steps based on the same circuit architecture to implement different logic functions compared to other logic approaches, as shown in Table 1 for XOR logic.
Figure 5. Demonstration board

We build a demonstration board based on FPGA and one 32x32 memristor crossbar array and test the perceptron for logic function. After training on PC through a perceptron model, we get the weights and set weights into the memristors with the form of a corresponding excitation voltage pulse. The results demonstrate that this scheme is effective.

4. Conclusion

In this paper, a neural network model and circuit based on resistance array, to implement linear separable logic functions such as OR, AND within perceptron based on Memristor-based neural networks and also to implement non-linear separable logic function such as XOR within multilayer perceptron based on Memristor-based neural networks. We could implement different logic functions in the same crossbar and circuit by adjusting the resistance of the memristor. It’s easily changed by tuning of the input voltage pulse on a memristor while keeping its resistance in the same for different logic operations. Compared with the state-of-the-art algorithm, the proposed method can tolerant the variations of crossbar arrays to a certain degree as weights are more robust while we trained the neural networks with a data set with variation simulating to real memristor device.

References

[1] I. Vourkas I et al. IEEE Circuits & Systems Magazine, 16(3):15-30,2016
[2] Gantz J, Reinsel D. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. IDC iView: IDC Analyze the future, 2012, 2007(2012): 1-16.
[3] J. Yang et al. Nature Nanotechnology. 8(1):13-24,2013
[4] Waldrop, M. M. The chips are down for Moore's law. Nature 530, 144-147 (2016).
[5] Wong, H.-S. P. & Salahuddin, S. J. N. n. Memory leads the way to better computing. 10, 191 (2015).
[6] Huang, P. et al. Reconfigurable Nonvolatile Logic Operations in Resistance Switching Crossbar Array for Large-Scale Circuits. Adv Mater 28, 9758-9764,2016.
[7] Gaillardon P E , Luca Amarú, Siemon A , et al. The Programmable Logic-in-Memory (PLiM) computer[C] 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2016.
[8] Hur, Rotem Ben, and S. Kvatinsky. "Memristive memory processing unit (MPU) controller for in-memory processing." Science of Electrical Engineering IEEE, 2017:1-5.
[9] Yang, J. J., D. B. Strukov, and D. R. Stewart. "Memristive devices for computing." Nature Nanotechnology 8.1(2013):13-24.
[10] Vourkas I, Sirakoulis G C. Emerging Memristor-Based Logic Circuit Design Approaches: A Review[J]. IEEE Circuits & Systems Magazine, 2016, 16(3):15-30.
[11] Zhong S, Elia A, Alessandro B, et al. Logic Computing with Stateful Neural Networks of Resistive Switches[J]. Advanced Materials, 2018:1802554.
[12] Chen, Bing, et al. "Efficient in-memory computing architecture based on crossbar arrays." Electron Devices Meeting IEEE, 2016:17.5.1-17.5.4.
[13] S. Balatti, S. Ambrogio, D. Ielmini, IEEE Trans. Electron Dev. 2015, 62, 1831.
[14] P. Huang, J. Kang, Y. Zhao, S. Chen, R. Han, Z. Zhou, Z. Chen, W. Ma, M. Li, L. Liu, X. Liu, Adv. Mater. 2016, 28, 9758
[15] S. Sun, J. Li, Z. Li, H. Liu, Q. Li, and H. Xu, “Low-consumption neuromorphic memristor architecture based on convolutional neural networks,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2018, pp. 1–6.
[16] S.-Y. Sun, Z. Li, J. Li, H. Liu, H. Liu, and Q. Li, “A memristor-based convolutional neural network with full parallelization architecture,” IEICE Electron. Express, vol. 16, no. 3, 2019, Art. No. 20181034.
[17] Sun, Z., Ambrosi, E., Bricalli, A. & Ielmini, D. J. A. M. Logic Computing with Stateful Neural Networks of Resistive Switches. 1802554 (2018).
[18] Xumeng Zhang, Wei Wang, Qi Liu. An Artificial Neuron Based on a Threshold Switching Memristor[J]. IEEE Electron Device Letters, 2018, 39(2):308-311.
[19] B. Li, S. Yi, H. Miao, W. Yu, Y. Chen, and H. Yang, “Memristor-based approximated computation,” in Proc. Int. Symp. Low Power Electron. Design, 2013, pp. 242–247