Training and Operation of Multi-layer Convolutional Neural Network Using Electronic Synapses

Yi Ding¹, Penglong Li¹, Jiaqi Liu³, Ding Luo¹, Xiaolong Li¹ and Zhenghao Li²

¹Chongqing Geomatics and Remote Sensing Center, Chongqing, China
²Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China
³College of Engineering and Applied Sciences, Nanjing University, Nanjing, Jiangsu, China
Email: dy@dl023.net; lpl@dl023.net; lizh@cigit.ac.cn

Abstract. For the reason that electrotonic-based memristive devices have been developing rapidly, memristive synapses show a strong superiority in being exploited to construct the neural network system. Nanoscale of memristive devices provides wide prospects for making the hardware implementation of neuromorphic networks. The primary neural network can be satisfactorily implemented on the memristor, which means that memristors can be applied to simple machine learning tasks. However, training and operation of the peculiar neural network with multilayer special processing functions on memristors is still a challenging problem. In this paper, we introduce the experimental implementation of transistor-free metal-oxide memristive crossbars, with device variability sufficiently low to allow operation of integrated neural network, in a multilayer convolutional neural network. Our network consists of multiple 3×3 memristive crossbar arrays both on the convolutional layers and the last layer, which reduces the challenge for the practical implementation of the deep networks. To perform the perfect recognition of the shape in the 27×27 pixel binary images, we bring forward a new coarse-grain variety of the gradient descent algorithm to train the proposed network. Finally, our trained network achieves desirable accuracy.

1. Introduction

At present, professional-level GPUs have already reduced a lot of pressure on the applications of artificial intelligence. However, using a processor under the traditional Von Neumann architecture to train large-scale neural networks not only consumes a lot of power, but also consume a lot of time because the data has to be folded back and forth between memory and processor.

The emergence of a new device allows the integration of computing and storage, which is memristor. The memristor is the fourth basic component that represents the relationship between magnetic flux and electric charge proposed by Chua in 1971. The memristor was first made by Hewlett-Packard in 2008 [1]. The memristor has the ability to memorize charge, which means that determining the resistance of the memristor can calculate the amount of charge flowing through the memristor. In addition, the Set, Reset, and Read voltage of memristor have totally different values. Among them, the Read voltage does not change the resistance of the memristor while reading the resistance value. It is this feature that enables memristors to integrate computing and storage.
At present, many researchers have conducted many studies on memristor-based neural network calculations and given many excellent results [2-11]. Prezioso et al. is one of the earliest teams that combined memristor and neural network calculations [2]. Among them, the neural network based on memristor is mostly the most basic full-connected layer, which limits the performance of the memristor. The traditional memristor array arrangement method greatly increases the difficulty and time for preparation as the size of the neural network expands, but also makes the hardware crosstalk noise more obvious, thereby affecting the accuracy. Even though Dong et al. innovatively implemented convolutional neural networks (CNNs) by placing resistor locations and differentiating the same array [9]. However, this kind of hardware convolution structure is not universal, and while copying a huge number of the same structures, it does not guarantee the complexity of the latter differential amplifier circuit.

In this paper, the original structure of the large memristor array is improved, by arranging the 3×3 basic memristor arrays in a matrix in a certain manner to construct the large-scale functional array. We created a physical 3×3 basic memristor array and made pulse training experiment on it, which proved that we can use pulse training to achieve the change in our required step size. In neural network, the process of convolution in convolutional neural networks is modified for the characteristics of the 3×3 basic memristor arrays. Finally, the built-up convolutional neural networks achieve 80% recognition rate for 10 handwritten digits.

2. Basic 3×3 Memristor Array
In this paper, small-scale memristor arrays are used as basic arrays and then arranged in a certain way to form large-scale functional arrays, which reduces the impact on precision, preparation difficulty, and time on hardware. Compared with the conventional memristor-based neural network implementation method, since the basic memristor array is arranged in a certain manner to form a large functional array, no collinearity or other connection is generated between the basic memristor arrays at all, so it is not susceptible to hardware noise. At the same time, the difficulty and time cost to develop an ultra-small basic memristor array is much smaller than that of directly implementing the function array according to the amount of information. The ultra-small basic memristor array can show stronger adaptability after the application is complicated. In addition, ultra-small memristor arrays with the same structure can be mass-replicated and fabricated, and the manufacturing process is mature.

Since the convolution kernel used in the convolutional neural network is 3×3, the basic memristor array size is also set to 3×3. The specific internal structure is shown in figure 1. We can see that the basic 3×3 memristor array adopts the 1R structure, which means that only one memristor is used to form the cell. Each bit line in the basic memristor array is grounded through an ammeter which measures the sum of all currents flowing through the three memristors. When reading the resistance and current of a single memristor at a specific location in the basic memristor array, the rest of the word lines except the word line of the memristor should be grounded.

3. Pulse Training on Basic Memristor Array
We use the material Pt/TaOx/TiN to fabricate a 3×3 basic memristor array, and record its current-voltage curve and pulse training process. Since the step size of the weight is set to 670Ω in the simulation training, it is necessary to prove that the step length can be implemented in hardware. Among them, the current-voltage curve is as shown in figure 2, and the curve shape is stable under the test of a plurality of cycles, which shows that the memristor array works in a stable state.

During the pulse training, Reset uses a pulse amplitude of -1V and Set uses a pulse amplitude of 0.85V, both of them use a pulse period of 2μs and a pulse duty ratio of 50%. Reset pulse training is shown in figure 3, and Set pulse training is shown in figure 4.

In Reset training, the conductance value changes linearly with the number of training pulses, which means that resistance values, as same as neural network weights, can be adjusted by applying a certain number of pulses. In figure 3, the resistance value is raised 91.6 ohms under 200 pulses. If you want to raise 670 ohms, the resistance value will change to 384.6 + 670 = 1054.6 ohms, at this time the
conductance value is 0.94ms. Based on the linear relationship between the conductance value and the number of pulses, it can finally be calculated that approximately 664 pulses are needed.

During the set training, the conductance value changes step by step about every 50 pulses. Even if the amount of conductance for each step change is only about 4μs, the change to the resistance can reach about 90 ohms, and as the conductance increases, the change range of resistance gradually decreases in the step. If you want to raise 670Ω, you need about 670/90 = 7.4 steps, which means you need about 372 pulses.

The pulse training curve illustrates that memristor resistance can be used as a weight, and directional fixed value adjustment of weight can be achieved. It also proves that using 670Ω as the step size of the adjustment weight can be achieved.

4. Experiments

4.1. Combine Memristor Array with CNNs
The comparison between the traditional CNNs and the proposed memristive CNNs is shown in table 1. First, the traditional convolutional neural network has a convolution step size of 1, and the proposed memristive convolution uses a convolution step of length 3. Second, conventional convolutional neural networks typically use the same convolution kernel to process all blocks of pixels, while the convolution
kernel used for each 3×3 pixel block of the memristive convolution is not the same. In addition, changes have been made to the pooling. The original pooling layer adopts downsampling, maximum value and so on to reduce the dimension of the feature. Memristor pooling directly reduces the dimension by taking the sum of image blocks, which is the simplest way to be implemented with a circuit. Finally, in order to ensure that the number of layers of the memristive neural network is shallow, the depth of the neural network is defined as two layers. The first layer is convolution+pooling layer, which is used for feature output. The second layer is fully connected layer to generate recognition results.

### Table 1. The comparison between traditional CNNs and memristive CNNs.

|                        | Traditional CNNs | Memristive CNNs |
|------------------------|------------------|-----------------|
| Convolution step       | Usually 1        | 3               |
| Convolution kernel     | 1                | Multiple        |
| Pooling                | General downsampling | Find the sum of the image blocks |
| Convolution + Pooling  | Separate         | Merge           |

#### 4.2. Experimental Details

The test dataset is MNIST database of handwritten digits, which stands for Modified National Institute of Standards and Technology database [12]. The MNIST database has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

The convolution + pooling layer memristor array is formed by a plurality of basic memristor arrays arranged in a 9×9 matrix. The fully connected memristor array consists of a plurality of basic memristor arrays arranged in a matrix of 27×4. Before the image is input to the memristor, set both the resistance of the convolution + pooling layer and the fully connected layer to ROFF. Finally, in the case where there are only two layers, the identification of 10 digits can be completed with an 80% recognition rate. It can be clearly seen from figure 5 that the use of gradient descent can increase the accuracy by about 5 percentage points.

**Figure 5.** The recognition results.

#### 5. Conclusion

In this paper, the recognition of 10 digits of a convolutional neural network based on memristor is realized, and the recognition accuracy is 80%. In the neural network construction, the memristive convolutional neural network consists of a convolution + pooling layer and a fully connected layer. Structurally, the small memristor array with the size of 3×3 is arranged as the basic array in different matrix manners into different large-scale functional arrays for each layer of our neural network. In the
future, we will further extend our research to the study of 3D convolutional neural networks, which will be an interesting and meaningful challenge.

Acknowledgements
This research was supported by the Open Project of Chongqing Geomatics and Remote Sensing Center (YDX2019Z104), the Chongqing Technology Innovation and Application Development Project (cstc2019jscx-msxmx0299) and the Chongqing Talent Plan (CQYC201903188).

References
[1] Snider G S 2008 Proc. IEEE International Symposium on Nanoscale Architectures p 85.
[2] Prezioso M, Merrikhbayat F, Hoskins B D, Adam G C, Likharev K K and Strukov D B 2015 Nature 521 61.
[3] Yao P, Wu H, Gao B, Eryilmaz S B, Huang X and Zhang W 2017 Nat. Commun. 8 15199.
[4] Hu L, Fu S, Chen Y H, Cao H, Liang L, Zhang H, Gao J, Wang J and Zhuge F 2017 Adv. Mater. 29 1606927.
[5] Pantazi A, Woźniak S, Tuma T and Eleftheriou E 2016 Nanotechnology 27 355205.
[6] Wu C, Kim T W, Choi H Y, Strukov D B and Yang J J 2017 Nat. Commun. 8 1.
[7] Choi S, Tan S H, Li Z, Kim Y, Choi C, Chen P Y, Yeon H, Yu S and Kim J 2018 Nat. Mater. 17 335.
[8] Serb A, Bill J, Khiat A, Berdan R, Legenstein R and Prodromakis T 2016 Nat. Commun. 7 12611.
[9] Dong Z, Zhou Z, Li Z F, Liu C, Jiang Y N and Huang P 2017 Silicon Nanoelectronics Workshop p 145.
[10] Chowdhury A P, Kulkarni P and Bojnordi M N 2018 J. Low Power Electron. Appl. 8 38.
[11] Yao P, Wu H, Gao B, Tang J, Zhang Q, Zhang W, Yang J J and Qian H 2020 Nature 577 641.
[12] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 Proceedings of the IEEE 86 2278.