Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 7
TPCEE 2024

Research on Memristor-Based Neural Networks

Haozheng Ling *

College of Physics, Donghua University, Shanghai, China

* Corresponding Author Email: Ling_HZ@mail.dhu.edu.cn

Abstract. Memristors, a new kind of resistive device, have attracted considerable attention in recent years because of their nonvolatile nature, low energy consumption, and highly integrated performance. The key theory of the memristor is described, and its application in the structure of neural networks and arithmetic optimization is discussed. First, the fundamental models and operating properties of the memristor are evaluated, with emphasis on its voltage, current, and time variations. This paper reviews the performance of a variety of models based on the concept of memory, which can be used to reduce energy consumption and improve computation efficiency. Additionally, advancements in memristor technology that enhance the scalability and reliability of neural network architectures are explored. An optimal combination of pruning and quantizing is presented, which can greatly decrease the amount of data required by the neural network. Results show that the memristor has great potential to generate high performance, low power, and high efficiency in artificial intelligence applications, making it a promising component in the future of computing technology

Keywords: Memristor; Neural Networks; Optimization Algorithms; Low Power Consumption; Efficient Computing.

1. Introduction

Memristors have attracted considerable interest for their neural network applications due to their distinctive ability to modulate resistance. These devices alter resistance based on electric charge and magnetic flux, with various models developed to accurately represent their behavior, including one that mirrors the synaptic behavior of recent memristive devices.

A scalable method for online gradient descent learning using memristors has been introduced for neural network training. This method utilizes an artificial synapse consisting of a single memristor to store the weight and two CMOS transistors for control, providing robustness and noise immunity akin to software implementations.

A significant advancement is the CKFO (Convolution Kernel First Operated) algorithm, enhancing CNN efficiency by enabling pruned weights to be used without modification. This greatly reduces parameters and multiplication operations while preserving high accuracy.

Moreover, techniques like Weight Noise Injection (WNI) mimic random fluctuations during memristor switching, aiding networks in adapting to non-ideal characteristics and maintaining performance. These innovations enable efficient, large-scale neural network implementations with memristor technology, decreasing power consumption and increasing robustness, thus expanding the applicability of artificial neural networks.

2. Memristor-Based Neural Networks

2.1. The Working Principle of Memristors

The control function may be represented by a charge $M(q) = d\Phi/dq$, (1)

where M(q) is the memristance (in Ω), ϕ is the magnetic flow, and q is the electric charge. Currently managed HP memristors [1-3].

$$v(t) = R(t)i(t) \tag{1}$$

$$\mathrm{R(t)} = \mathrm{RON}\ \omega(\mathrm{t})/\mathrm{D} + \mathrm{ROFF}(1 - \omega(\mathrm{t})/\mathrm{D}) \tag{2}$$

In this scenario, ω(t) represents the width of the dopant region, initially defined as ω(0) = ω0 ∈ R. The memristance, denoted as R(t), is given by R(t) = M(q). Here, D stands for the thickness of the titania layer [4]. RON signifies the low internal resistance when the memristor is fully doped (ω(t) = D), while ROFF indicates the high internal resistance when the memristor is entirely undoped (ω(t) = 0). Both current and voltage are expressed accordingly. To account for the unique properties of memristors, various memristor models are compared [5-7]. A novel model is introduced, aligning with the synaptic behavior observed in newly implanted memristive devices [8]. The derivative of the state variable in this proposed memristive model is

$$\frac{d\omega(t)}{dt} = \begin{cases} \mu_v \frac{R_{ON}}{D} \frac{i_{off}}{i(t)-i_0} f\big(\omega(t)\big), v(t) > V_{T+} > 0 \\ 0, V_{T-} \le v(t) \le V_{T+} \\ \mu_v \frac{R_{ON}}{D} \frac{i(t)}{i_{on}} f\big(\omega(t)\big), v(t) < V_{T-} < 0 \end{cases} \tag{3}$$

In this example, VT +, V- and V- are constants, μv is mean ion mobility, and VT + and V- are both positive and negative thresholds [4]. There is a correlation between voltage variation and impulse count (time), which means that conductivity is proportional to the number of pulses used [9]. It is also possible to add an a nonlinear ion drift phenomenon, for example a reduction in the ion drift rate close to the bounds, with a window.

The nature of the SBT memristor is inherently nonlinear. This model demonstrates the voltage threshold characteristics of the SBT memristor. When the voltage applied exceeds the positive threshold, the memristor transitions from a high-resistance state to a low-resistance state. Conversely, if the voltage falls below the negative threshold, the memristor reverts to its high-resistance state.

2.2. Neural Networks Based on Memristors

Mei Guo and colleagues introduce an innovative approach to enhance the performance of hierarchical neural networks. This technique leverages local update rules facilitated by emerging memristor technology. The method relies on an artificial synapse where the synaptic weight is stored in a single memristor, complemented by two CMOS transistors. Results indicate that the proposed synapses exhibit greater robustness and reliability compared to their software-based counterparts.

Such circuits can be used for deep learning and deep learning in multiple layers of neural networks. Compared to current CMOS-only models, the proposed circuitry offers the possibility of massively parallel parallel processing in one IC with hundreds of thousands of self-adapting synapses in one IC [10].

Mei Guo et al. proposed a SBT-memristor-based framework for deep convolutional neural networks, which is based on the combination of pruning and quantization. Their framework includes a set of storage, activation, and max-pooling, which can be used to perform highly accurate tasks.

Additionally, they introduced a novel method to enhance the performance of deep convolutional neural networks. It was discovered that an SBT-memristor-based convolutional neural network could reduce the model size by 85.7%, lower the energy dissipation by 72.9%, and decrease the memristor count by 80.96% [11]. For SBT-memristor-based convolutional neural networks, the detection rate achieved is 95.67%, although the accuracy stands at only 3.62%. Furthermore, the MNIST dataset benefits from a higher identification rate and reduced energy consumption. This approach has accelerated the adoption of artificial neural networks across various applications. Moving forward, enhanced algorithms will be employed in the design of memristor-based neural networks.

2.3. Research on Algorithms for Memristor-Based Neural Networks

Shuai Dong et al. present a BP-GA algorithm that uses real memristors to train MNNs. Originally used in the SLP based on the membrane, BP-GA was tested for the capability and operating properties of the device. It achieves accuracies exceeding 85% across a wide range of learning rates ($\eta = 0.01 – 50$), highlighting its robustness to variations in learning rate. BP-GA has also attained a high precision (> 85%) in the presence of a considerable number of memristor nonideal conditions, such as big nonlinearity ($\nu > 6$), finite conductivity ($n = 3$), various ON/OFF ratios (across three orders of magnitude), and significant noise (as much as 40%). This flexibility is attributable to the operating mechanism of BP-GA, which allows for significant adjustment of the weights with no change in their characteristics as a result of the cumulative gradient. This flexibility allows for effective adjustment of weights even with small η or large Gmax/Gmin ratios. The performance of BP-GA has also been verified on MLP (MNIST data set) and CNN (CIFAR-10 data set), which has shown that it can be used in many computer vision tasks. Thus, the BP-GA algorithm suggested can make it easier to implement MNNs in hardware with real membranes [12].

This paper introduces a novel convolution algorithm called Convolution Kernel First Operation (CKFO). The proposed method addresses the issue of no practical reduction in computation, even when the weights of a convolutional neural network are pruned. This paperprovide an implementation of a Convolutional Neural Network (CNN) using an ex situ approach, where the CNN is initially trained in TensorFlow and subsequently transformed into Simulink for model evaluation. At last, the validity of this model is proved. Next, CNN is trimmed and retrained, and then the simulated model is modified based on the trimmed parameters. Furthermore, this paperhave shown that the convolutional neural network is able to generate the same data as the previous one, and it is not necessary to modify it. In other CNNs, this is usually difficult because of the uneven distribution of trimmed weights. Using this method, this papercan decrease the parameters by 75.24%, and multiply times in convolution layer by 30.1%, and only 0.06% less precision.

Memristor crossbars can be used to calculate large quantities of data in a number of ways. However, due to structural changes, resistance changes may be seen in the postmemristor programming, resulting in a reduction in the accuracy of the trained network. This paperhave observed that when the mem is programmed to have a higher resistance state (representing logical 0) compared with a lower resistance state (representing logical 1), there is a marked difference in resistance. Shuai Dong and colleagues propose an innovative approach called two-phase weight mapping and memristor programming. In the initial stage, high-value, untrimmed weights are mapped onto the crossbar. Due to the high number of zero weights in the sparse network, most memristor scans are programmed to the highest resistance, thereby achieving excellent performance. In this phase, Shuai Dong et al. also adjust some of the zero weights to small, non-zero values. Mapping these low values ensures that the memristor is more resistant to change, allowing it to effectively compensate for variations in the high-value weights. Experimental results demonstrate that this approach can effectively address the problem using memristors and that its precision matches that of a trained software network [13].

3. A BP Training Circuit

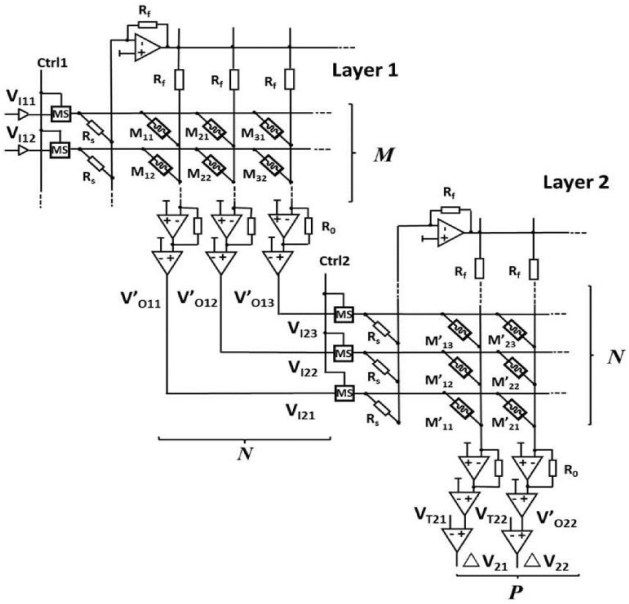

Figure 1. Proposed memristor-based MNN [4].

Figure 1 illustrates the realization of BP training in NN circuitry [4].

There are four stages in the education process [14]:

Firstly, the NN is trained by applying an input voltage to the first layer, and then registering the output error of the neurons in the second layer. Then, by means of the second-tier weights, they are backpropagated, and an error is recorded in Tier 1. Next, the synaptic weights of the second layer are updated according to the errors from the second layer. Finally, the synaptic weights of the first layer are updated according to the errors in the first layer.

The four steps of the proposed BP training are as follows:

First, in the course of training, the MS is enabled by setting the input signals Ctrl1 and Ctrl2 to VH, and an input voltage is supplied to the first level of the neurons. Then, a comparison is made between the output error and the desired one, and an error term is produced and amplified so that the membrane can be adjusted. In Step 2, Ctrl1 and Ctrl2 are left at VH for back-propagation of Tier 2 errors, which are then applied to Tier 2 weights to produce Tier 1 errors. As for Step 3, Ctrl1 and Ctrl2 are set to zero so that the 2nd level cross-bar is cut off from the first level. Finally, a training unit generates a training pulse, which is used to improve the synaptic weights in the second layer. Finally, in Step 4, a similar procedure is performed to update the synaptic weights in the Tier 1 memristor cross-wires when Ctrl1 and Ctrl2 remain at 0 (Figure 2).

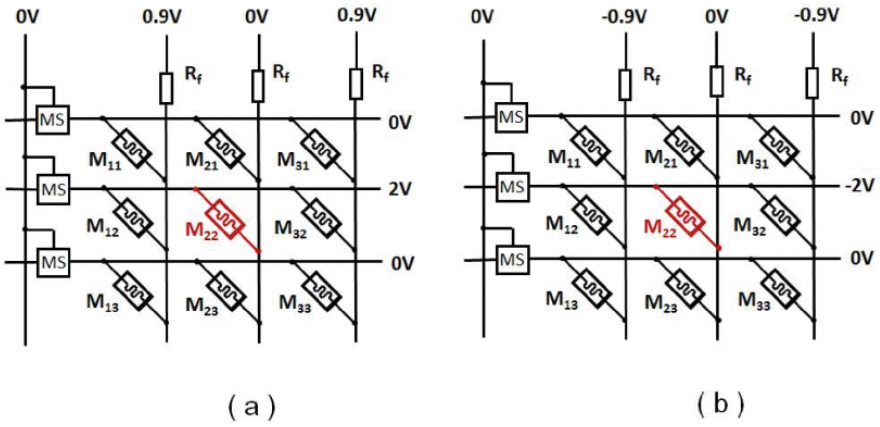

Figure 2. Synaptic balance based on memory. (a) the adjustment of M22 from 1 to -1, and (b) the adjustment of M22 from -1 to 1 [4].

Drawing inspiration from the architecture and theory of the human brain, Spiking Neural Networks (SNNs) have gained significant traction in the realm of computer vision over recent years. The shift towards hardware implementation of artificial neural networks is evident, despite numerous existing challenges. Memristors, known for their rapid programming capabilities, low energy consumption, and compatibility with CMOS technology, have emerged as promising candidates for such hardware realizations. This paper commences with an exploration of the basic principles underlying SNNs, followed by a detailed discussion on hardware-based SNN techniques. Furthermore, the integration of custom-optimized algorithms is proposed to enhance the performance and energy efficiency of SNNs.

In this paper, an algorithm named Weight Noise Injection (WNI) is proposed, which can be used to simulate the stochastic oscillation of the memristors in conduction switching. This noise is due to the randomicity of the memristor, which makes it possible to accommodate the suboptimal properties of the memristors in the process of training. The precision of the net can be increased by 10 percent when the change in MNIST is 40%, and the standard distance between high-low-conductivity and low-conductivity-ratio is 0.5. Nevertheless, there was a smaller increase in PeMS data (3.3%) because of the limitations of the scope for improvement.

This inherent randomness can also be effectively utilized. For example, Parami Wijesinghe and his team combine a stochastic memristor with various circuit components (like capacitors and transistors) to create an impulse neuron featuring random activation capabilities. When integrated with the memristor array, this circuit unit can be incorporated into a deep stochastic SNN. To enhance the random activation of a memristor, they have developed a technique to measure each memristor's potential and current, ensuring that every memristor neuron functions correctly while minimizing energy consumption.

The results show that the accuracy of all memristor deep random SNN is equivalent to that of ANN, but compared with Complementary Metal-Oxide-Semiconductor (CMOS) hardware, this method can save about 6.4 times of power.

The Memristor Array performs several critical functions: it executes the vector-matrix multiplication of weights and inputs, simulates neuron dynamics within a nerve computing unit, calculates the error between target and actual outputs in a learning unit, and generates modulated pulses through an impulse modulator to update each cell state. To enhance the efficiency of STM circuits, the authors suggested a method inspired by a single neuron computing model from [15]. The deployment of neural networks in circuits poses significant challenges due to the complexity of the circuitry and the interactions among its components. Consequently, the self-learning neural circuitry discussed is implemented with the on-line minimum Average Square (LMS) algorithm [16]. Additionally, section IV-A describes memristor synapses as pairs of interconnected membranes, which adjust the synapse based on the conduction and excitability of the membrane, as detailed in [17-19]. The entire process is completed using PSpice. To validate our proposed approach, this paperemploy a memristor-based spiking neural network to recognize handwritten characters.

3.1. Step i:

The initialization stage includes the configuration of the network parameters and the matrix of the memristor. Each column of the array initializes memristors to a low conductivity through voltage pulses.

3.2. Step ii:

Pulse Injection: Every 28x28pixel digital MNIST picture is coded in the form of a pulse series (Vinput, i), which is injected into the ten neurons of 0-9.

3.3. Step iii:

Weight planning. Target weights shall be programmed by applying a variable number pulses for each column in accordance with the coded gray scale.

3.4. Step iv:

Error Computation: The error is calculated as the mean of the absolute difference of the input and output voltage

$$error = \frac{1}{n}\sum_{i=1}^{i=n}\left|V_{input,i} - V_{output,i}\right| \quad (4)$$

If the error converges to a sufficiently small value (previously set as 10^−5 in this study), proceed to Step v; otherwise, return to Step iii.

3.5. Step v:

Identification policy: Calculate the output pulses produced by every neuron, and decide the final identification of the most number of neurons.

4. Conclusion

As a novel electronic device, the memristor is becoming an essential component of neural network hardware implementations due to its unique memory characteristics and superior performance. This paper systematically introduces the working principles of memristors and explores memristor-based neural network design and optimization methods. Memristors not only significantly reduce the energy consumption and footprint of neural networks but also enable more efficient computation through their non-linear and threshold properties.

In the design of memristor-based neural networks, this paperprovide detailed explanations of the implementation methods for multilayer neural networks and convolutional neural networks. By integrating real-world applications, this paperdemonstrate their significant advantages in enhancing computational efficiency and reducing power consumption. Additionally, by introducing advanced optimization algorithms such as BP-GA and pruning quantization methods, this paperfurther enhance the performance and adaptability of neural networks.

In the future, with the continuous development and maturation of memristor technology, this paperhave reason to believe that memristor-based neural networks will find extensive applications in various fields, offering new solutions for efficient and low-power artificial intelligence systems. Continued research and exploration of the working mechanisms and optimization methods of memristors will contribute to further advancement in this field, bringing more innovation and breakthroughs to artificial intelligence hardware implementations.

References

[1] L. Chua, Memristor-The missing circuit element, IEEE Trans. Circuit Theory 18 (1971) 507-519.

[2] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams, The missing memristor found, Nature 453 (2008) 80-83.

[3] S. P. Adhikari, M. P. Sah, H. Kim, L. O. Chua, Three fingerprints of memristor, IEEE Trans. Circuits Syst. I Reg. Papers 60 (2013) 3008-3021.

[4] Y. Zhang, X. Wang, E. G. Friedman, Memristor-Based Circuit Design for Multilayer Neural Networks, IEEE Trans. Circuits Syst. I Reg. Papers 65 (2018) 677-686.

[5] S. Kvatinsky, E. G. Friedman, A. Kolodny, U. C. Weiser, TEAM: Threshold adaptive memristor model, IEEE Trans. Circuits Syst. I Reg. Papers 60 (2013) 211-221.

[6] S. Kvatinsky, M. Ramadan, E. G. Friedman, A. Kolodny, VTEAM: A general model for voltage-controlled memristors, IEEE Trans. Circuits Syst. II Express Briefs 62 (2015) 786-790.

[7] Ascoli, F. Corinto, V. Senger, R. Tetzlaff, Memristor model comparison, IEEE Circuits Syst. Mag. 13 (2013) 89-105.

[8] Y. Zhang, X. Wang, Y. Li, E. G. Friedman, Memristive model for synaptic circuits, IEEE Trans. Circuits Syst. II Exp. Briefs 64 (2017) 767-771.

[9] Y. Li et al., Activity-dependent synaptic plasticity of a chalcogenide electronic synapse for neuromorphic systems, Sci. Rep. 4 (2014) 4906.

[10] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, S. Kvatinsky, Memristor-Based Multilayer Neural Networks With Online Gradient Descent Training, IEEE Trans. Neural Netw. Learn. Syst. 26 (2015) 2408-2421.

[11] M. Guo, Y. Sun, Y. Zhu et al., Pruning and quantization algorithm with applications in memristor-based convolutional neural network, Cogn. Neurodyn. 18 (2024) 233-245.

[12] S. Dong, Y. Chen, Z. Fan, K. Chen, M. Qin, M. Zeng, X. Lu, G. Zhou, X. Gao, J.-M. Liu, A backpropagation with gradient accumulation algorithm capable of tolerating memristor non-idealities for training memristive neural networks, Neurocomputing 494 (2022) 89-103.

[13] S. Jin, S. Pei, Y. Wang, A variation tolerant scheme for memristor crossbar based neural network designs via two-phase weight mapping and memristor programming, Future Gener. Comput. Syst. 106 (2020) 270-276.

[14] R. Hasan, T. M. Taha, Enabling back propagation training of memristor crossbar neuromorphic processors, Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2014, pp. 21-28.

[15] M. Hu et al., Memristor-based analog computation and neural network classification with a dot product engine, Adv. Mater. 30 (2018).

[16] Suresh et al., Realizing spike-timing dependent plasticity learning rule in Pt/Cu:ZnO/Nb: STO memristors for implementing single spike based denoising autoencoder, J. Micromech. Microeng. 29 (2019).

[17] Z. Wang, X. Wang, Z. Zeng, Memristive circuit design of brain-like emotional learning and generation, IEEE Trans. Cybern., Jul. 2021.

[18] Z. Liao, J. Fu, J. Wang, Ameliorate performance of memristor-based ANNs in edge computing, IEEE Trans. Comput. 70 (2021) 1299-1310.

[19] Z. Wang et al., Self-activation neural network based on self-selective memory device with rectified multilevel states, IEEE Trans. Electron Devices 67 (2020) 4166-4171.