# Memristor-based neural network circuit with weighted sum simultaneous perturbation training and its applications

Cong Xu[a], Chunhua Wang[a,*], Yichuang Sun[b], Qinghui Hong[a], Quanli Deng[a], Haowen Chen[a]

[a]*College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China*
[b]*School of Engineering and Technology, University of Hertfordshire, Hatfield AL10 9AB, U.K.*

**Abstract**

In this work, a full circuit of memristor-based neural network with weighted sum simultaneous perturbation training is proposed. Firstly, a synaptic circuit is designed by using a pair of memristors, which can represent negative, zero, and positive synaptic weights. Secondly, a full circuit of the neural network is designed, with all operations being completed on the circuit without any computer aid. The neural network is trained with the weighted sum simultaneous perturbation algorithm. The algorithm does not involve complex derivative calculation and error back propagation, and it only applies perturbations to weighted sum, so the circuit implementation is more simple. Finally, application simulations of the proposed neural network circuit are performed via PSpice. The results of simulation indicate that the memristor-based neural network is practical and effective.

*Key words:* Memristor, neural network, synaptic weight, circuit design, recognition, weighted sum simultaneous perturbation.

## 1. Introduction

The neural network is one of the most important branches of artificial intelligence, and the implementation of neural networks in hardware is a research hotspot. HP labs physically realized the memristor in 2008 [1]. Meanwhile, the application of memristors is more and more widely seen in recent years, such as memristive chaotic circuit [2–7], memristive neural network [8–14]. As a nonvolatile programmable resistor, memristor has the advantages of high density, low power, and good scalability. It is the most promising candidate for realizing storage of synaptic weight in artificial neural network. So the research in memristor-based neural network is gaining more and more attention.

In memristor-based neural network circuits, memristance is applied to represent synaptic weight variation. In [15–17], the synaptic operation was realized by the memristor circuit, which greatly simplified the structure of the Hopfield neural network, and the proposed neural network was proved effective via simulation results. Multi-layer neural network is applicable to solve the problem of nonlinear classification. Some multi-layer neural networks based on memristor were proposed by using memristor to simulate weight [18–20]. These papers applied multiple neural networks to pattern recognition. Hu et al. [21] proposed a simple dynamic synapse based on memristor, and studied the spiking neural network by utilizing the dynamic synapses. In [22],

*Corresponding author at: College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China

*Email addresses:* wch1227164@hnu.edu.cn (Chunhua Wang )

the memristance of synaptic circuit can be updated by the positive voltage, and the circuits of memristor-based neural network were proposed for recognition and classification. Besides, some other kinds of neural networks were realized by memristor, such as perceptron [23], recurrent neural network [24], RBF neural network [25],[26], cellular neural network [27],[28], pulse-coupled neural network [29],[30], echo state network [31], and convolutional neural networks [32–34].

As we all know, an efficient training system is critical for weight adjustment in neural networks. Researchers proposed neural networks based on different learning algorithms in recent years. Ref. [35] proposed a neural network based on WTA and STDP learning rules, which was applied to position detection. In [36], a memristor-based neural network with LMS algorithm was proposed. Ref. [37] realized a simple architecture of synapse by using CMOS transistor and memristor, and the online gradient descent training algorithm was used in the process of training. Besides, the error backpropagation (BP) is the most popular training algorithm in the neural network, which was proposed by Rumelhart [38]. In [39], a modified BP algorithm was implemented in multi-layer neural network by using memristor-based synapse. Ref. [40] proposed a training system which based on backpropagation rule for adjustment of synaptic weight. Besides, some hardware-friendly algorithm were used in circuit implementation of neural network. For instance, the memristor-based neural network with the random weight change (RWC) algorithm was proposed [41], where synaptic weights were turned by a small constant randomly. The design simplified the circuit structure. Wang et al. [42] proposed a new circuit architecture of neural network based on weight simultaneous perturbation algorithm (WSP), which significantly simplified the circuit of weight adjustment and allowed simpler and easier circuit implementation of the neural network. However, for the BP algorithm, the error propagation and derivative of nonlinear active function are needed and the calculation of derivative is difficult to be realized by an analog circuit. These characteristics make the circuit structure of neural network complex. The RWC algorithm does not have complex operations, but the value of error variance is not utilized in algorithm by only considering the sign of error variation, which results in low convergent speed. In WSP algorithm, the value and sign of the error variation are used to the adjustment of weights. However, perturbations are added to all weights simultaneously, which makes the circuit of the multi-layer neural network not concise.

To solve the above problems, a full circuit of memristor-based neural network with weighted sum simultaneous perturbation training is proposed. The weighted sum simultaneous perturbation (WSSP) algorithm is a simple and effective training method for neural network. In the neural network with WSSP algorithm, a perturbation signal is added to the weighted sum, then the weights updated based on the difference between unperturbed and perturbed error function. Different from the BP algorithm, the WSSP algorithm does not involve the complex derivative calculation and error back propagation. And compared with the WSP algorithm, the WSSP algorithm only applies perturbations to weighted sum, thus the circuit implementations are more concise. In this paper, a synaptic circuit is designed firstly by using a pair of memristors, which can achieve negative, zero, and positive synaptic weights. Second, a full circuit of neural network based on WSSP algorithm is designed, the training rule and weight adjustment are implemented in circuits without any computer aid. The WSSP algorithm updates weight by calculating the difference between unperturbed and perturbed error, this method utilizes the value and the sign of the error variation, which overcomes the defect of the RWC algorithm. Besides, the computational structure of WSSP algorithm is much simpler by contrast with the BP algorithm and WSP algorithm, which make the circuit operations much more concise and feasible. Finally, odd parity problem and face recognition are performed by the proposed memristor-based neural network via PSpice simulation. The effectiveness and practicability of the circuit is proved by simulation results.

The remainder of this paper is organized as follows. The basic background on the weighted

sum simultaneous perturbation algorithm is given in Section 2. The circuit of neuron and neural network is described in Section 3. In Section 4, the simulation results of the neural network circuit for odd parity and face recognition are analyzed. Section 5 presents the conclusion drawn from this work.

## 2. Background

### *2.1. Weighted sum simultaneous perturbation algorithm*

The perturbation algorithm [43] was suggested as an alternative to back-propagation. Compared with the BP algorithm, there is no complex derivative calculation and error back propagation in this algorithm, so it is easy to be implemented in hardware. In WSP algorithm [42], the perturbation p is applied to all weights simultaneously, but in the circuit implementation, the complexity of circuit will increase with the increased number of synaptic weight. Different from WSP algorithm, the perturbation p is only applied to weighted sum in WSSP algorithm, which makes the circuit design more concise.

In the neural network, the output y and the error function E are defined as

$$y = f(\sum_{i=1}^{n} w_i x_i) \tag{1}$$

$$E = \|t - y\|^2 \tag{2}$$

where $x_i$ is the input of the network, $w_i$ is the weight of the network, $f$ is the nonlinear activation function, and $t$ is the target value of output.

The detailed flowchart of WSSP algorithm is shown in Fig. 1. A small positive constant perturbation signal $p$ is added to the weighted sum $s$, and the perturbation affects the output and error. The detailed calculation process of WSSP algorithm as follows. First, the error without a perturbation is calculated. Second, a perturbation signal is applied to the weighted sum, and the error with a perturbation is calculated. Finally, The weight updated according to the difference between unperturbed and perturbed error function. As a consequence, the weight update rule is given by

$$\begin{cases} \Delta w_i = -\alpha \frac{\partial E}{\partial w_i} = -\alpha \frac{\partial E}{\partial s} \frac{\partial s}{\partial w_i} = -\alpha \frac{\partial E}{\partial s} x_i \\ w_i(m+1) = w_i(m) + \Delta w_i \end{cases} \tag{3}$$

$$\begin{cases} \Delta w_i \approx -\alpha \frac{\Delta E}{\Delta s} x_i = -\alpha \frac{\Delta E}{p} x_i \\ w_i(m+1) = w_i(m) + \Delta w_i \end{cases} \tag{4}$$

where

$$\Delta E = E_{per} - E \tag{5}$$

and $\alpha$ is the learning rate, $p$ is a positive constant, $E_{per}$ is the error with a perturbation, $E$ is the error without a perturbation.

### *2.2. MNN architecture with WSSP algorithm*

Multi-layer neural network (MNN) is applicable to solve the problem of nonlinear classification. The architecture of MNN trained by WSSP algorithm is shown in Fig. 2. The MNN cascades multiple single-layer neural networks usually, and the mathematical model of a neural network with two layers is defined as

$$Y = W_2 f(W_1 X) \tag{6}$$

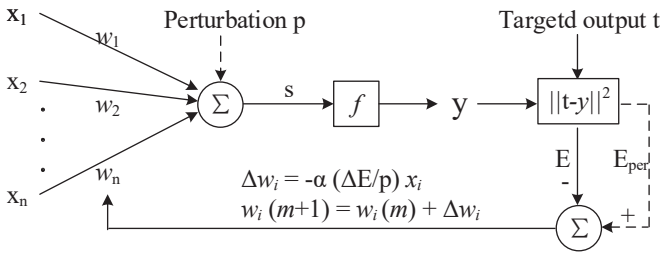



**Fig. 1.** The flowchart of WSSP algorithm.

where $f$ denotes the nonlinear activation function, $W_1$ and $W_2$ are the weight matrix. As described as Eq. (4), the weight adjust rule for network with two layers is defined as

$$\begin{cases} \Delta \mathrm{W}_1 = -\alpha \frac{\Delta E}{p} \mathrm{X}_1 \\ \Delta \mathrm{W}_2 = -\alpha \frac{\Delta E}{p} \mathrm{X}_2 \end{cases} \tag{7}$$

where $X_1$ and $X_2$ are input of the two-layer neural network, respectively. As for a general MNN with WSSP algorithm, all weight matrices are updated according to the following formula synchronously.

$$\Delta \mathrm{W}_k = -\alpha \frac{\Delta E}{p} \mathrm{X}_k \tag{8}$$

where $k$ is the index of the neural network layer.

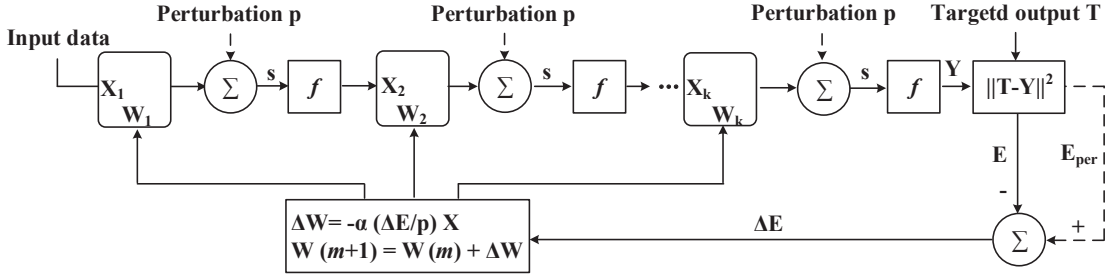



**Fig. 2.** The architecture of MNN trained by WSSP algorithm

### *2.3. Algorithms comparison*

In BP algorithm, the error propagate from output layer to the hidden layers, and weights adjustment are related to the derivative calculation of the activation function. WSSP algorithm does not contain a complex derivative calculation and error back propagation. Besides, compared with the WSP algorithm, the WSSP algorithm only applies perturbations to the weighted sum as shown in Fig. 2, which simplifies the algorithm structure and implementation complexity. To validate the effectiveness of the algorithm, comparison experiments were conducted using two typical algorithms and the WSSP algorithm in Matlab (R2016b). A multilayer neural network with three training algorithms performed recognition on the MNIST data set of handwritten digits. The neural network consists of 38 input × 80 hidden × 10 output. 5000 images were used for training and 1000 images for testing in our experiment. The network was trained by WSSP, BP, and WSP algorithms, respectively. The mean squared errors vs the number of iterations for WSSP, BP, and WSP algorithms are shown in Fig. 3. From the curve of training error, it indicates that the convergence speed of WSSP is faster than BP and WSP algorithms. So the WSSP algorithm is feasible and effective.

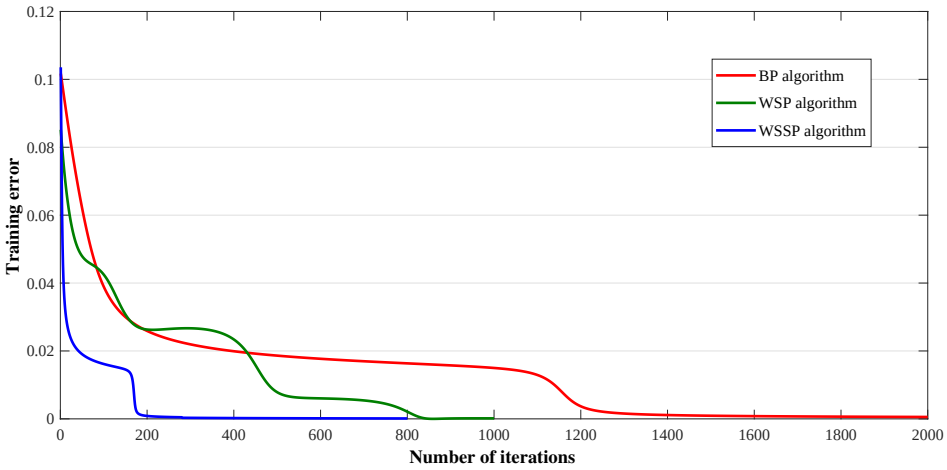



**Fig. 3.** The curve of training error.

## 3. Circuit design and simulations

### *3.1. Memristor*

As the fourth basic circuit element, memristor has the same regulation mechanism as the weight regulation of synapse, so the memristor is the most promising device for designing circuits of neural network. In this work, the circuit of neural network was designed using a voltage-controlled memristor model [44]. If the applied input signal exceeds the threshold, the memristance of this device will change, else it remains unchanged. The expressions of the memristor model are described as

$$R(t) = R_{on}\frac{\omega(t)}{D} + R_{off}(1 - \frac{\omega(t)}{D}) \tag{9}$$

$$\frac{d\omega(t)}{dt} = \begin{cases} \mu_v \frac{R_{on}}{D}\frac{i_{off}}{i_{on}} f(\omega(t)), & v(t) < V_{T-} < 0 \\ 0, & V_{T-} \le v(t) \le V_{T+} \\ \mu_v \frac{R_{on}}{D}\frac{i_{off}}{i(t)-i_0} f(\omega(t)), & v(t) > V_{T+} > 0 \end{cases} \tag{10}$$

where $\omega(t)$ denotes the width of the doped region, $\mu_v$ denotes the average ion mobility, $i_0$, $i_{off}$, and $i_{on}$ are constants, $V_{T-}$ and $V_{T+}$ are negative and positive threshold voltages, respectively, and the window function is

$$f(\omega(t)) = 1 - (\frac{2\omega(t)}{D} - 1)^{2a} \tag{11}$$

where a is a positive integer parameter of window function. The main parameters used in this work are $R_{on} = 100\Omega, R_{off} = 10k\Omega, D = 10nm, \mu_v = 1\times10^{-12}m^2s^{-1}\Omega^{-1}, i_{on} = 1A, i_{off} = 1\times10^{-5}A, i_0 = 1\times10^{-3}A, V_{T+} = 2V, V_{T-} = -2V, a = 4$.

### *3.2. Neuron circuit*

The designed neuron circuit is shown in Fig. 4. A synapse in the circuit is composed of two memristors, and the activation function is implemented by using amplifiers and resistors. The output $V_o$ of the neuron is given by

$$\begin{aligned} V_o &= R_f\left[(I_{1+} - I_{1-}) + (I_{2+} - I_{2-}) + \cdots + (I_{n+} - I_{n-})\right] \\ &= R_f[V_1(\tfrac{1}{M_{1a}} - \tfrac{1}{M_{1b}}) + \cdots V_n(\tfrac{1}{M_{na}} - \tfrac{1}{M_{nb}})] \\ &= R_f \sum_{i=1}^{n} V_i w_i \end{aligned} \tag{12}$$

where

$$w_i = (\frac{1}{M_{ia}} - \frac{1}{M_{ib}})=\mathrm{G}_{ia} - G_{ib} \tag{13}$$

When the saturation voltage of amplifier is $V_{ss}$ and $V_{cc}$, it can be also described as

$$V_o = \begin{cases} V_{ss}, & V_o < V_{ss} \\ R_f \sum_{i=1}^{n} V_i w_\mathrm{i}, & V_{ss} \le V_o \le V_{cc} \\ V_{cc}, & V_o > V_{cc} \end{cases} \tag{14}$$

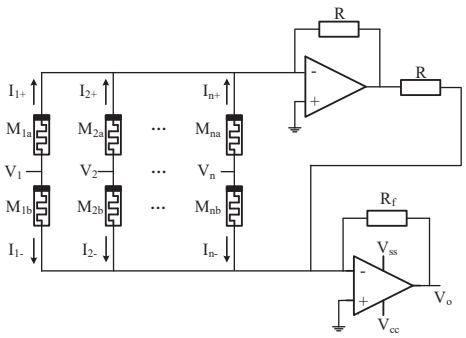



**Fig. 4.** The memristor-based neuron circuit

PSpice simulation of memristor synapse is performed, the results are shown in Fig. 5. The polarity of memristor $M_{ia}$ is clearly opposite to that of $M_{ib}$, so the change in memristances of $M_{ia}$ and $M_{ib}$ is opposite under a positive or a negative voltage. According to Eq. (13) and Fig. 5(b), it indicates that the synapse can achieve positive, zero, and negative synaptic weights.

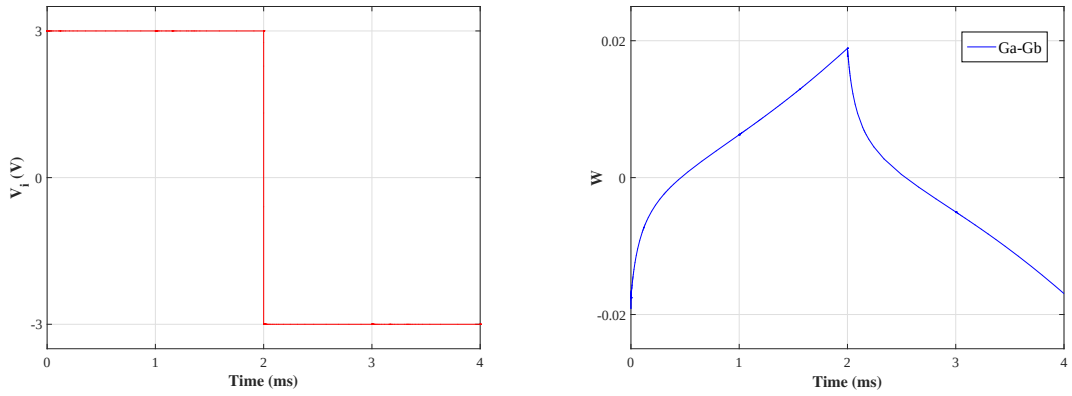



**Fig. 5.** The weight adjustment of memristor synapse. (a) Input voltage $V_i$. (b) Time variation of weight W.

### *3.3. Circuit design of neural network with WSSP training algorithm*

In off-chip neural network training works, the training process is implemented in software, then the calculated weights are downloaded to the circuit, which leads to the lack of the parallelism of neural network. As shown in Fig. 6, for the full circuit implementation of neural network, a full circuit is designed by using the WSSP algorithm. Different from the commonly used BP algorithm, the WSSP algorithm does not involve the complex derivative calculation, this makes the circuit implementation is more concise. The training is divided into two processes, forward

propagation and feedback adjustment process, and they are executed alternately, controlled by signal $V_c$ in the circuit, until the algorithm converges. The switching function is performed by transmission gate switches SW, the used operational amplifier is LM741, the sample and hold is LF398. The main parameters $R = 3k\Omega, R_f = 600\Omega, V_{cc} = 15V, V_{ss} = -15V, V_{T+} = 2V, V_{T-} = -2V$. The details of the full circuit are described as follows.

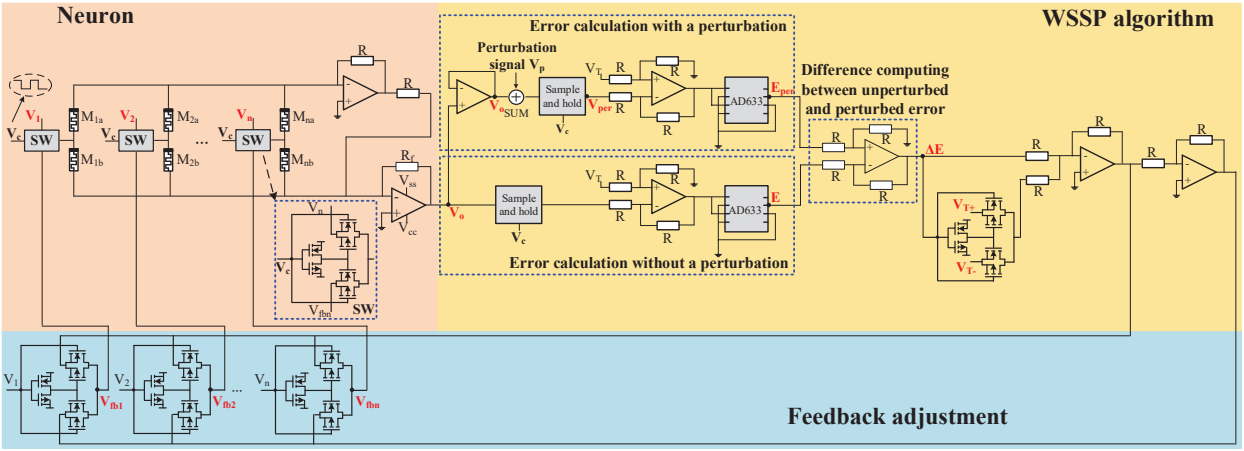



**Fig. 6.** Circuit of the memristor-based neural network with WSSP algorithm.

a. The process of forward propagation: When the control signal $V_c$ is at a high level, the input signals input to the network for error calculation. The input signal of the neural network is denoted as $V_i (i = 1, 2, ..., n)$, $V_p$ is a small positive voltage, which is the perturbation signal of WSSP algorithm. $V_o$ is the output without a perturbation, $V_{per}$ is the output with a perturbation, $V_T$ is the target value. The error $E$ and $E_{per}$ are calculated by

$$E = (V_T - V_o)^2 \tag{15}$$

$$E_{\text{per}} = \left(V_T - V_{per}\right)^2 \tag{16}$$

Then the difference between unperturbed and perturbed error is calculated by

$$\Delta E = (E_{\text{per}} - E) \tag{17}$$

b. The process of feedback adjustment: When the control signal $V_c$ is at the low level, the adjusting voltages $V_{fbi} (i = 1, 2, ..., n)$ feed back to network for weight updating. In the WSSP algorithm, as shown in Eq. (4), the weight adjustment depends on the error difference, perturbation signal, and input. In the circuit design, the adjusting signal of synapse weight depends on the error difference and the sign of input signal. Besides, we add the threshold voltage to ensure the adjusting voltage is greater than threshold of memristor, it can be described as

$$V_{\text{fb}i} = \begin{cases} -(\Delta E + V_{T-}) \cdot sign(V_i), if(\Delta E) < 0 \\ -(\Delta E + V_{T+}) \cdot sign(V_i), if(\Delta E) > 0 \end{cases} \tag{18}$$

where $V_{T-}$ and $V_{T+}$ are the negative and positive threshold of memristor. If the adjusting voltage is greater than threshold of memristor, the memristance will increase or decrease, otherwise it remains unchanged. That is, the synaptic weight increases or decreases until the error is close to zero.

Table 1: Weight update situation.

| | $\Delta E$ | sign($V_i$) | $V_{fbi}$ | $M_{ai}$ | $M_{bi}$ | Weight update |
|---|---|---|---|---|---|---|
| Case 1 | + | + | - | Increase | Decrease | Decrease |
| Case 2 | + | - | + | Decrease | Increase | Increase |
| Case 3 | - | + | + | Decrease | Increase | Increase |
| Case 4 | - | - | - | Increase | Decrease | Decrease |

### *3.4. Synaptic weight update*

The threshold memristor is used in our work, if the adjusting voltage is greater than the threshold of memristor, the memristance will increases or decreases, otherwise it remains unchanged. Based on the rule of the WSSP algorithm, the four possible weight update situations during the training process as listed in Table 1, and the simulation results of case 1 and case 2 are shown in Fig. 7. As can be seen from the results, the memristor-based synaptic weight can be adjusted by WSSP algorithm effectively.

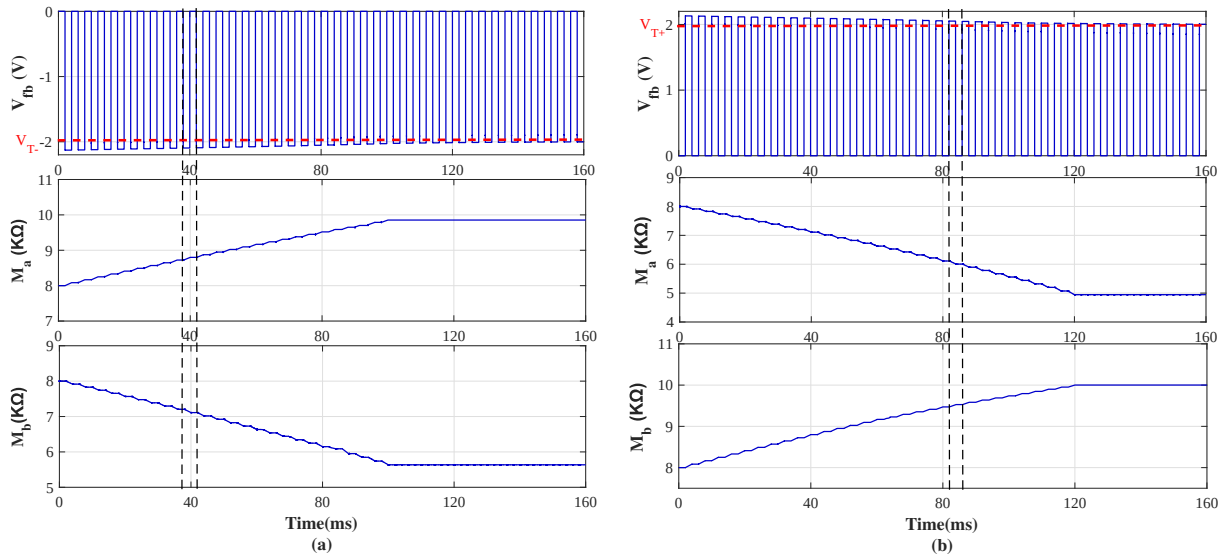



**Fig. 7.** The process of synaptic weight update. (a) The variation of $V_{fb}$, $M_a$ and $M_b$ in case 1. (b) The variation of $V_{fb}$, $M_a$ and $M_b$ in case 2.

Case 1: Input $V_i$ and error $\Delta E$ are positive, adjusting voltage $V_{fb}$ is negative as shown in Table 1. As shown in Fig. 7 (a), the adjusting voltage $V_{fb}$ decreases gradually, since the actual output is close to the targeted output gradually. The memristance of $M_a$ increases, while the $M_b$ decreases. As described in Section 3.3, a cycle of training is divided into two stages: adjustment and forward propagation. Take one cycle as an example, in adjustment stage (38-40ms), the memristance of $M_a$ and $M_b$ increases and decreases by adjusting voltage, respectively, and in forward propagation stage (40-42ms), the error difference is recalculated, the memristances remain unchanged in this stage. Synaptic weights constantly adjusted until the error is close to zero. About at 100ms, the $V_{fb}$ is equal to threshold of memristor, the training process of weight is completed, then the memristances remain unchanged.

Case 2: As shown in Fig. 7 (b), the positive adjusting voltage $V_{fb}$ decreases gradually, the memristance of $M_a$ decreases, and $M_b$ increases, so the synaptic weight increases in this case. For instance, in one cycle (82-86ms), the memristances of $M_a$ and $M_b$ are adjusted based on $V_{fb}$ in first half cycle, then the error is recalculated in the second half cycle. The training process is completed until the error is close to zero. In this simulation, the training process is completed about at 120ms, then the memristances remain unchanged.

Case 3: The error $\Delta E$ is negative, while the sign of input is positive, so the $V_{fb}$ is a positive voltage as listed in Table 1. The update process of weight is similar to that of case 2.

Case 4. In this case, input and error are negative voltages, so the adjusting voltage is negative. It is similar to case 1, the adjustment of synaptic weight is a decrease process in this case.

## 4. Applications

The proposed memristor-based neural network circuit with WSSP training algorithm dose not involve the complex derivative calculation and error back propagation. The full circuit architecture of memristor-based multilayer neural network (MNN) is shown in Fig. 8, and the details of the error calculation circuit by WSSP algorithm is shown in Fig. 6. For verifying the practicability of the circuit, the memristor-based multi-layer neural network is applied to parity problem and face recognition.

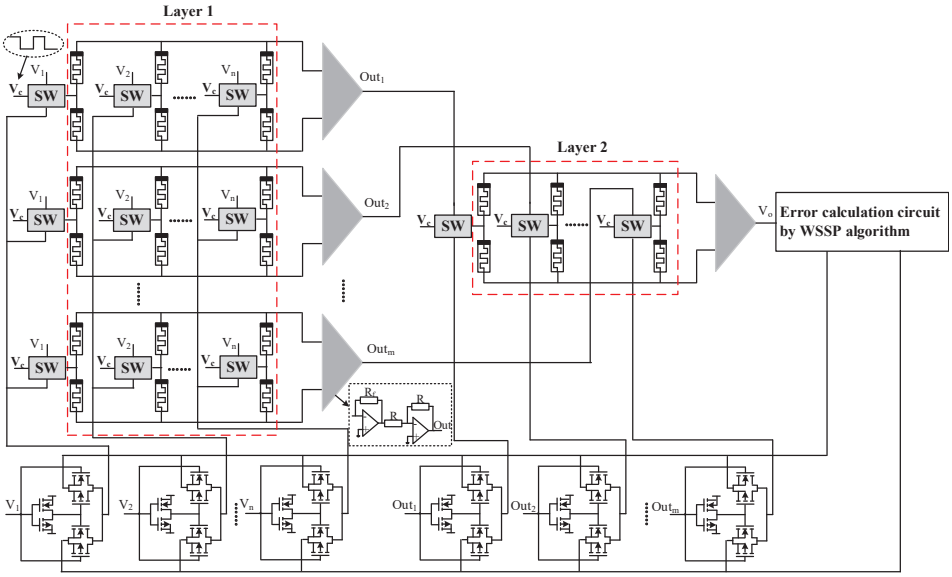
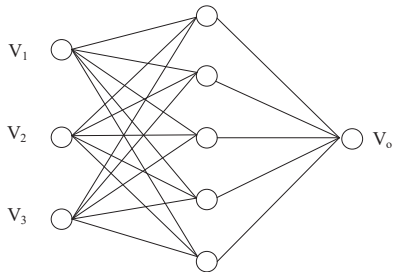



**Fig. 8.** The full circuit architecture of memristor-based MNN with WSSP algorithm.

### *4.1. Parity problem*

A parity check is a method to check the correctness of code transmission. To prove the feasibility of the WSSP algorithm in MNN, the memristor-based MNN is adopted to the three input odd parity problem. The simulation was implemented by using a network of 3 inputs × 5 hidden × 1 output is shown in Fig. 9(a). The truth table of 3-bit odd parity is shown in Fig. 9(b). The circuit of memristor-based MNN for odd parity problem is composed of 3 inputs × 5 hidden × 1 outputs. Besides, the weight adjustment of the input and hidden layer depends on the sign of the input signals and the error difference as shown in Eq. (7). It can be seen that there is no complex derivative calculation and error back propagation in the circuit. So, the architecture of the MNN with WSSP algorithm is simpler.

Four different input case of odd parity were simulated by PSpice. The voltages -1V and 1V represent logic '0' and logic '1' in the circuit simulation, respectively. Fig. 10(a-c) are the input voltages, the four different cases were (-1,-1,-1)V, (-1,-1,1)V, (1,1,-1)V and (1,1,1)V, and the corresponding target output were -1V, 1V, -1V and 1V. The four cases were input to the network iteratively. Fig. 10(d) is the output of the MNN, the output voltage is approaching to the target value during the training process. After 64ms, the output is equal to the target value. The simulation results indicated that the memristor-based MNN can make parity check correctly after training.




(a)

| $V_1$ | $V_2$ | $V_3$ | $V_o$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

(b)

**Fig. 9.** Odd Parity. (a) Multilayer neural network. (b) Truth table.

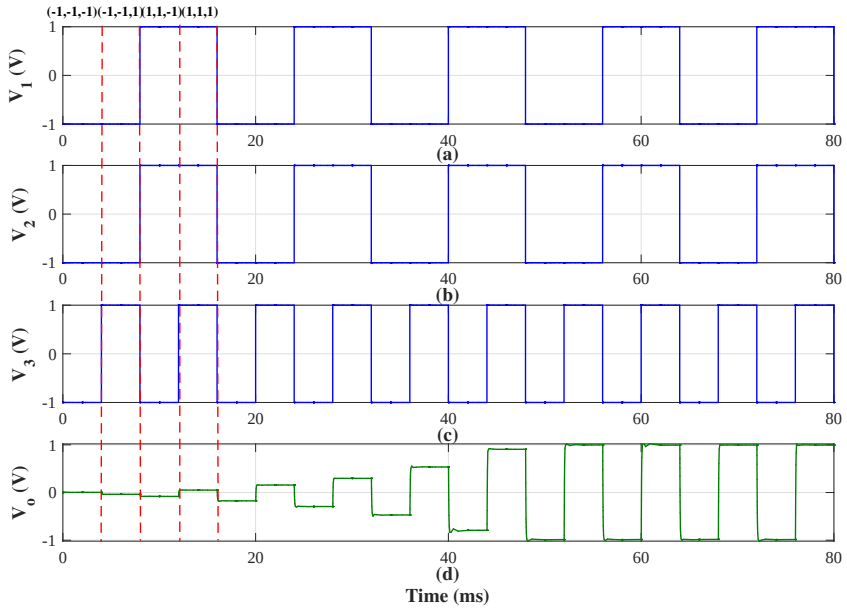



**Fig. 10.** The simulation results of Parity problem. (a) The input voltage $V_1$. (b) The input voltage $V_2$. (c) The input voltage $V_3$. (d) The output voltage $V_o$.

### *4.2. Face recognition*

Face recognition plays an important role in daily life, and it is widely used in intelligent systems, such as automatic picture archiving system, identity verification system and so on. The proposed memristor-based neural network with WSSP algorithm was applied to face recognition by using ORL database of faces. The flowchart of face recognition is shown in Fig. 11. In order to reduce the complexity of data, the feature extraction and dimensionality reduction of images are processed at first, then the reduced image feature vector are normalized to (-1,1). For circuit simulation, the feature values are converted to voltage values as the inputs of network. Based on the size of feature vector and the number of classes, the size of the memristor-based MNN is set as $16 \times 10 \times 6$. One people with ten different images are shown in Fig. 12, the first 7 images are used for training and the last 3 images are used for testing. The target value of this kind of face images is set as [0.5V,-0.5V,-0.5V,-0.5V,-0.5V,-0.5V]. Besides, the images preprocessing were implemented in Matlab, and the training process of the proposed memristor-based neural network was simulated by PSpice.

The PSpice simulation result is shown in Fig. 13. As we can seen, during the training process (0-60ms), the feature vector of testing image as the inputs data, and the outputs are approaching to the target values gradually. After the training completed, the remaining 3 testing images as

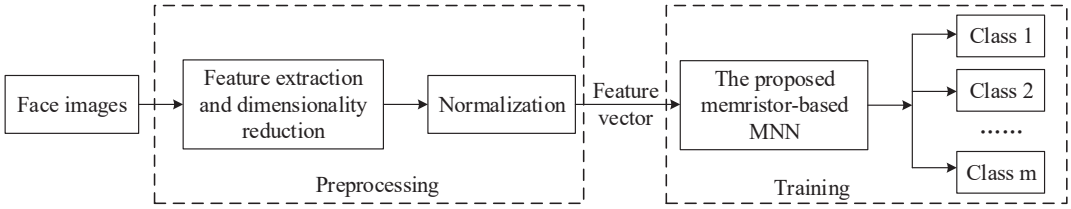



**Fig. 11.** The flowchart of face recognition based on memritor-based MNN.

**Fig. 12.** Face images for recognition: the first 7 images are used for training and the last 3 images are used for testing.

the input of the network, the images were recognized correctly. The training and testing results indicated that the proposed memristor-based MNN with WSSP algorithm is effective for face recognition.

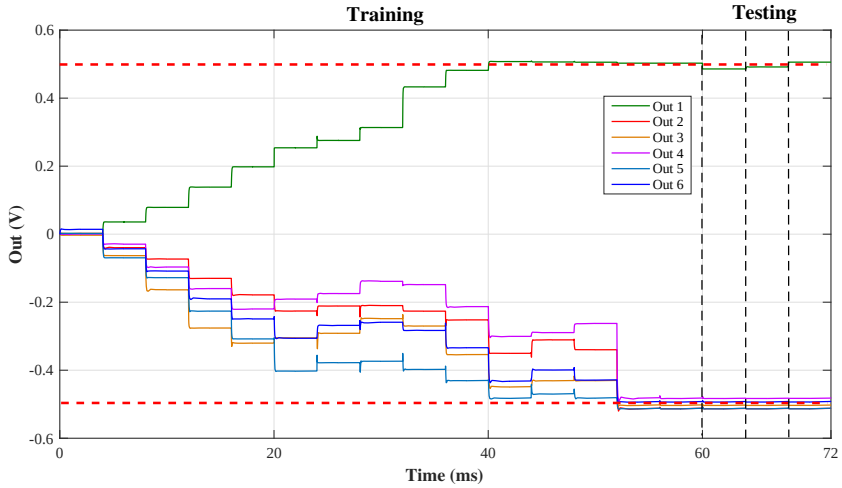



**Fig. 13.** The PSpice simulation results of face recognition: the output voltages of the memristor-based MNN.

## 5. Conclusion

In this paper, a full memristive circuit of neural network with the weighted sum simultaneous perturbation algorithm is proposed. The weighted sum simultaneous perturbation training algorithm does not involve the complex derivative calculation and error back propagation, and it only applies perturbations to the weighted sum, which makes the circuit implementation more concise and feasible. By using a pair of memristors, the synaptic circuit can represent negative, zero, and positive synaptic weights. Based on the memristive synaptic circuit, a circuit of neural network with WSSP algorithm is presented. The training process of the memristor-based neural network contains forward propagation and feedback adjustment, which are implemented in the circuit without computer aid. Besides, the four possible weight update situations during the training process are analyzed in detail. Finally, the memristive circuit of the neural network is applied to the odd parity problem and face recognition.

The simulation results verified the effectiveness of the proposed memristive circuit in practical application. However, the proposed memristive circuit adopts discrete components to realize the training process, which is inconvenient for large-scale integration. To accomplish the widespread application of memristor-based neural networks, we will focus on designing a large-scale integrated memristive circuit with simple structure and low power consumption in future work.

## Acknowledgments

This work is supported by the Major Research Plan of the National Natural Science Foundation of China (No.91964108), the National Natural Science Foundation of China (No.61971185) and Natural Science Foundation of Hunan Province (2020JJ4218).

## References

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

[2] Q. Zhao, C. Wang, X. Zhang, A universal emulator for memristor, memcapacitor, and meminductor and its chaotic circuit, Chaos 29 (1) (2019) 013141.

[3] F. Yu, L. Li, B. He, L. Liu, S. Qian, Z. Zhang, H. Shen, S. Cai, Y. Li, Pseudorandom number generator based on a 5D hyperchaotic four-wing memristive system and its FPGA implementation, Eur. Phys. J.-Spec. Top.(2021), doi:https://doi.org/10.1140/epjs/s11734-021-00132-x.

[4] W. Xie, C. Wang, H. Lin, A fractional-order multistable locally active memristor and its chaotic system with transient transition, state jump, Nonlinear Dyn. (2021) 1–19.

[5] H. Lin, C. Wang, Y. Sun, W. Yao, Firing multistability in a locally active memristive neuron model, Nonlinear Dyn. 100 (4) (2020) 3667–3683.

[6] M. Zhu, C. Wang, Q. Deng, Q. Hong, Locally active memristor with three coexisting pinched hysteresis loops and its emulator circuit, Int. J. Bifurcation Chaos 30 (13) (2020) 2050184.

[7] H. Lin, C. Wang, W. Yao, Y. Tan, Chaotic dynamics in a neural network with different types of external stimuli, Commun. Nonlinear Sci. Numer. Simul. 90 (2020) 105390.

[8] Q. Hong, Z. Shi, J. Sun, S. Du, Memristive self-learning logic circuit with application to encoder and decoder, Neural Comput. Appl. 33 (2021) 4901–4913.

[9] H. Lin, C. Wang, Q. Hong, Y. Sun, A multi-stable memristor and its application in a neural network, IEEE Trans. Circuits Syst. II-Express Briefs 67 (12) (2020) 3472–3476.

[10] W. Yao, C. Wang, Y. Sun, C. Zhou, H. Lin, Exponential multistability of memristive cohen-grossberg neural networks with stochastic parameter perturbations, Appl. Math. Comput. 386 (2020) 125483.

[11] W. Yao, C. Wang, J. Cao, Y. Sun, C. Zhou, Hybrid multisynchronization of coupled multistable memristive neural networks with time delays, Neurocomputing 363 (2019) 281–294.

[12] Q. Hong, R. Yan, C. Wang, J. Sun, Memristive circuit implementation of biological nonassociative learning mechanism and its applications, IEEE Trans. Biomed. Circuits Syst. 14 (5) (2020) 1036–1050.

[13] C. Zhou, C. Wang, Y. Sun, W. Yao, Weighted sum synchronization of memristive coupled neural networks, Neurocomputing 403 (2020) 211–223.

[14] W. Yao, C. Wang, Y. Sun, C. Zhou, H. Lin, Synchronization of inertial memristive neural networks with time-varying delays via static or dynamic event-triggered control, Neurocomputing 404 (2020) 367–380.

[15] J. Yang, L. Wang, Y. Wang, T. Guo, A novel memristive hopfield neural network with application in associative memory, Neurocomputing 227 (2017) 142–148.

[16] Q. Hong, Y. Li, X. Wang, Memristive continuous hopfield neural network circuit for image restoration, Neural Comput. Appl. 32 (3) (2020) 8175–8185.

[17] C. Chen, J. Chen, H. Bao, M. Chen, B. Bocheng, Coexisting multi-stable patterns in memristor synapse-coupled hopfield neural network with two neurons, Nonlinear Dyn. 95 (4) (2019) 3385–3399.

[18] Y. Zhang, X. Wang, E. G. Friedman, Memristor-based circuit design for multilayer neural networks, IEEE Trans. Circuits Syst. I-Regul. Pap. 65 (2) (2018) 677–686.

[19] H. Raqibul, M. T. Tarek, Y. Chris, A fast training method for memristor crossbar based multi-layer neural networks, Analog Integr. Circuits Process. 93 (3) (2017) 443–454.

[20] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, J. Strachan, M. Barnell, Q. Wu, R. Williams, J. Yang, Q. Xia, Efficient and self-adaptive in-situ learning in multilayer memristor neural networks, Nat. Commun. 9 (1) (2018) 2385.

[21] M. Hu, Y. Chen, J. J. Yang, W. Yu, H. L. Hai, A compact memristor-based dynamic synapse for spiking neural networks, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 36 (8) (2017) 1353–1366.
[22] L. Yang, Z. Zeng, X. Shi, A memristor-based neural network circuit with synchronous weight adjustment, Neurocomputing 363 (2019) 114–124.
[23] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, D. Strukov, Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521 (7550) (2015) 61–64.
[24] T. Guo, L. Wang, M. Zhou, S. Duan, A multi-layer memristive recurrent neural network for solving static and dynamic image associative memory, Neurocomputing 334 (2019) 35–43.
[25] T. Li, S. Duan, J. Liu, L. Wang, T. Huang, A spintronic memristor-based neural network with radial basis function for robotic manipulator control implementation, IEEE Trans. Syst. Man Cybern. -Syst. 46 (4) (2016) 582–588.
[26] T. Li, S. Duan, J. Liu, L. Wang, An improved design of RBF neural network control algorithm based on spintronic memristor crossbar array, Neural Comput. Appl. 30 (6) (2018) 1939–1946.
[27] X. Hu, G. Feng, S. Duan, L. Liu, A memristive multilayer cellular neural network with applications to image processing, IEEE Trans. Neural Netw. Learn. Syst. 28 (8) (2017) 1889–1901.
[28] E. Bilotta, P. Pantano, S. Vena, Speeding up cellular neural network processing ability by embodying memristors, IEEE Trans. Neural Netw. Learn. Syst. 28 (5) (2017) 1228–1232.
[29] S. Zhu, L. Wang, S. Duan, Memristive pulse coupled neural network with applications in medical image processing, Neurocomputing 227 (2017) 149–157.
[30] X. Xie, S. Wen, Z. Zeng, T. Huang, Memristor-based circuit implementation of pulse-coupled neural network with dynamical threshold generators, Neurocomputing 284 (2018) 10–16.
[31] S. Wen, R. Hu, Y. Yang, T. Huang, Z. Zeng, Y. Song, Memristor-based echo state network with online least mean square, IEEE Trans. Syst. Man Cybern. -Syst. 49 (9) (2019) 1787–1796.
[32] X. Zeng, S. Wen, Z. Zeng, T. Huang, Design of memristor-based image convolution calculation in convolutional neural network, Neural Comput. Appl. 30 (2) (2018) 503–508.
[33] Y. Peng, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. Yang, H. Qian, Fully hardware-implemented memristor convolutional neural network, Nature 577 (7792) (2020) 641–646.
[34] Y. Zhang, M. Cui, Y. Liu, L. Shen, Hybrid CMOS-memristive convolutional computation for on-chip learning, Neurocomputing 355 (25) (2019) 48–56.
[35] I. E. Ebong, P. Mazumder, CMOS and memristor-based neural network design for position detection, Proc. IEEE 100 (6) (2012) 2050–2060.
[36] S. Xiao, X. Xie, S. Wen, Z. Zeng, T. Huang, J. Jiang, GST-memristor-based online learning neural networks, Neurocomputing 272 (2018) 677–682.
[37] 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 (10) (2015) 2408–2421.
[38] D. E. Rummelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, Parallel Distributed Processing: Explorations in the Microstructure of Cognition 323 (2) (1986) 318–362.
[39] Y. Zhang, Y. Li, X. Wang, Synaptic characteristics of Ag/AgInSbTe/Ta-based memristor for pattern recognition applications, IEEE Trans. Electron Devices 64 (4) (2017) 1806–1811.
[40] R. Hasan, T. M. Taha, C. Yakopcic, On-chip training of memristor crossbar based multi-layer neural networks, Microelectron. J. 66 (2017) 31–40.
[41] S. P. Adhikari, H. Kim, R. K. Budhathoki, C. Yang, L. O. Chua, A circuit-based learning architecture for multilayer neural networks with memristor bridge synapses, IEEE Trans. Circuits Syst. I-Regul. Pap. 62 (1) (2015) 215–223.
[42] C. Wang, L. Xiong, J. Sun, W. Yao, Memristor-based neural networks with weight simultaneous perturbation training, Nonlinear Dyn. 95 (4) (2019) 2893–2906.
[43] B. Widrow, M. A. Lehr, 30 years of adaptive neural networks: perceptron, madaline, and backpropagation, Proc. IEEE 78 (9) (1990) 1415–1442.
[44] Y. Zhang, X. Wang, Y. Li, E. G. Friedman, Memristive model for synaptic circuits, IEEE Trans. Circuits Syst. II Express Briefs 64 (7) (2017) 767–771.