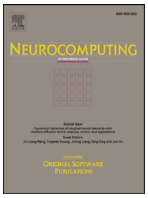
Neurocomputing 363 (2019) 114–124

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



A memristor-based neural network circuit with synchronous weight adjustment

Le Yang[a,b], Zhigang Zeng[a,b,*], Xinming Shi[a,b]

[a] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
[b] Key Laboratory of Image Processing and Intelligent Control of Education Ministry of China, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 4 November 2018
Revised 8 May 2019
Accepted 22 June 2019
Available online 18 July 2019

Communicated by Prof. Duan Shukai

Keywords:
Memristor-based neural network circuit
Memristor
Synaptic circuit
Neuron circuit
Synchronous weight adjustment

ABSTRACT

Memristor-based neural network circuits are considered as promising hardware implementations for artificial neural networks. In the training of memristor-based neural networks, it slows down the training speed that synaptic weights cannot be adjusted synchronously. Meanwhile, the relation between synaptic weight variation and the duration of control signals is unknown in neuron circuits, which is disadvantageous for the training. This paper proposes a neuron circuit connecting memristor-CMOS hybrid synaptic circuits whose memristance can be adjusted simply by the positive voltage. According to the structure of the neuron circuit, an inference is implemented to obtain the relation between the increment or decrement of synaptic weight and the lasting time of control signal. Then, based on the presented neuron circuit, this paper proposes a single layer neural network (SNN) achieving character recognition and a multi-layer neural network (MNN) for pattern classification. Under the control of a microcontroller, the two networks realize synchronous weight adjustment that all the synaptic circuits required to change can adjust memristance at the same time in an iteration.

1. Introduction

Memristor-based neural network circuits depend on the relation between current and voltage in hardware implementation to achieve computing acceleration. Hence, memristor-based neural network circuits are considered as a potential candidate for the implementation of artificial neural network [1–4].

The theoretical memristor was put forward by Chua [5], and Hewlett-Packard laboratories manufactured the first physical memristor in 2008 [6]. Since then, memristor has been applied to different aspects [7–10]. Memristor is a two terminal device whose memristance is usually changed by positive and negative voltages [11–13]. In [14], four MOS transistors and a complementary resistive switch constitute a memory cell in which the memristance is changed by the positive voltage. Only applying positive voltage to adjust memristance not only simplifies the design of power supplies, but also makes the control circuit easy to design.

In memristor-based neural network circuits, dynamical changing memristance is applied to represent synaptic weight variation [15–18]. In order to construct a memristor-based neural network circuit, we should design the basic units-synaptic circuit and neuron circuit at first. In [19,20], four memristors make up a bridge circuit to act as synaptic circuit in a neuron circuit. Then, based on the neuron circuit, a MNN circuit is designed to accomplish the face pose identification. However, due to the limitation of the bridge circuit, the variation of the synaptic weight is restricted in a small range (-1,1).

In memristor-based neural network circuits to accelerate computation, synaptic circuits are supposed to represent negative, zero, and positive weight. There are two cross-bar arrays with the same structure proposed in [21–24], in which the two memristors at the same position serve as a synaptic circuit. The same input signals are applied to the two cross-bar arrays to obtain outputs, and then the outputs make subtraction to map the difference of the two memristors' memristance as negative, zero, or positive weight. In [21–23], the memristor-based neural network circuits are used to realize character recognition.

Utilizing the same method, but with the different structure, the synaptic circuit in [25–27] is devised at one cross-bar array. When the differential input signals are applied to two rows of a cross-bar array to get outputs, the summation of the outputs maps the difference of memristance of two memristor as negative, zero, or positive weight. The circuit structure of the memristor-based neural network circuits in [26,27] does not match back propagation algorithm. Therefore, the algorithm is simplified to train the networks.

[*] Corresponding author at: School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China.
E-mail addresses: zgzeng@hust.edu.cn, zgzeng@mail.hust.edu.cn (Z. Zeng).

https://doi.org/10.1016/j.neucom.2019.06.048
0925-2312/© 2019 Published by Elsevier B.V.

In [28,29], the synaptic circuits consist of one memristor-based cross-bar array and additional resistors. The difference between the memristance of one memristor and the resistance represents negative, zero, or positive weight. Since the circuit structure of the memristor-based neural networks does not match back propagation algorithm, the algorithm is simplified to train the network to realize character recognition. However, for the memristor-based neural network circuits in [28,29], in an iteration, only one synaptic circuit can adjust memristance at a time. Hence, if n synaptic circuits need to be adjusted in an iteration, the iteration is accomplished with n times adjustments. This slows down training speed.

The synaptic circuits in [30–33] are composed of two transistors and one memristor at a cross-bar array. In [30,32] the memristor-based neural network circuits are utilized to realize iris classification. However, there is only one cross-bar array acting as the synaptic circuits. Therefore, when a synaptic weight changes to negative or zero value, the corresponding memristance is mapped to negative or zero value. The memristor in these circuits cannot be implemented by physical passive memristors.

In majority of the existing memristor-based neural network circuits, there are two common deficiencies. On one hand, the relation between synaptic weight change and the lasting time of control signals is unknown in neuron circuits, which is disadvantageous for the training of memristor-based neural network circuits. On the other hand, the synaptic circuits cannot be adjusted synchronously. In an iteration, the synaptic circuits required to adjust cannot change at the same time. There are only a part of the synaptic circuits adjusting the corresponding memristance at a time. Hence, the iteration needs to adjust several times, which slows down training speed.

This paper proposes a neuron circuit which is compatible with CMOS logic level, so the synaptic weight can be adjusted simply by the positive voltage with relatively simple control circuit. According to the circuit structure of the neuron circuit, the relation between the synaptic weight variation and the last time of control signal is inferred in the paper. For the synaptic circuit, the variation range of synaptic weight can be varied by changing the corresponding resistance. In an iteration, the synaptic circuits can be adjusted synchronously. All the synaptic circuits can be adjusted at the same time, so the corresponding memristance is set to aimed values at one time to accomplish the iteration. This makes the training of the memristor-based neural network circuits efficient.

The remainder of this paper proceeds as follow: Section 2 introduces the design of neuron circuit. Section 3 presents the memristor-based neural network circuit for character recognition. Section 4 presents the synchronous weight adjustment in pattern classification network circuit. Conclusions are drawn in Section 5.

2. Circuit implementation of the neuron circuit

2.1. Neuron circuit connecting one synaptic circuit

The physical memristor manufactured by Hewlett-Packard laboratories is a $P_t/TiO_{2-x}/P_t$ structure in which doped TiO_2 and undoped TiO_2 are fabricated between two P_t electrodes. When a voltage is applied to the memristor, the ion motion causes the memristance to change. The memristor model is given by

$$v(t) = \left(R_L\frac{w(t)}{D} + R_H\left(1 - \frac{w(t)}{D}\right)\right)i(t) \tag{1}$$

$$\frac{dw(t)}{dt} = \frac{\mu_v R_L}{D}i(t) \tag{2}$$

Thus, $w(t)$ can be represented as

$$w(t) = \frac{\mu_v R_L}{D}q(t) \tag{3}$$

Since $R_L \ll R_H$ and $\frac{w(t)}{D} \leq 1$, the memristance variation can be given in a simplified way [6]

$$R_m(t) = R_H\left(1 - \frac{\mu_v R_L}{D^2}q(t)\right) \tag{4}$$

where D indicates the length of the two layers TiO_2, μ_v is the average ion mobility rate in the memristor, $q(t)$ is the total charge flowing through the memristor, $w(t)$ is the length of doped region, R_H and R_L represent the maximum and minimum memristance, respectively.

Synaptic circuit and neuron circuit are basic units in a memristor-based neural network circuit. The proposed neuron circuit which connects one synaptic circuit is shown in Fig. 1.

In Fig. 1, V_{I1} and V_{C1} represent the input signal and control signal, which are digital logic levels. The voltages of *logic* 0 and *logic* 1 in the paper are arranged as 0V and 5V, respectively. T_1 and T_2 are p-MOS transistors while T_3 and T_4 are n-MOS transistors. The memristor R_m, the four MOS transistors, the resistor R_{N1} and the operational amplifier A_1 constitute an inverting amplifier. Owing to *virtual ground*, the voltages at non-inverting terminal and inverting terminal of A_1 are both 0V [34,35]. When V_{I1} is *logic* 0, the voltage across the memristor R_m is 0V. No matter the logic level of V_{c1} is *logic* 0 or *logic* 1, there is no current flowing through R_m. Therefore, the memristance keeps unchanged in this condition.

If V_{I1}=1, V_{c1}=1, the inverter U_1 outputs *logic* 0. Thus, T_1 and T_3 turn on while T_2 and T_4 turn off. In this state, the current flows from V_{I1} through T_1, R_m, T_3 and R_{N1} to the output terminal of A_1, driving the memristance to increase. If V_{I1}=1, V_{c1}=0, T_2 and T_4 turn on while T_1 and T_3 turn off. In this situation, the current flows from V_{I1} through T_2, R_m, T_4 and R_{N1} to the output terminal of A_1, making the memristance reduce. The analysis demonstrates that the synaptic circuit simply uses the positive voltage to change the memristance and the neuron circuit is compatible with digital logic level. When V_{I1}=1, the output voltage of A_1 is a negative value

$$V_1(t) = -\frac{V_{I1}R_{N1}}{R_m(t)} \tag{5}$$

The resistors R_1, R_{N2} and operational amplifier A_2 also make up an inverting amplifier. The output voltage of A_2 is given by

$$V_2(t) = -\frac{V_{I1}R_{N2}}{R_1} \tag{6}$$

$V_1(t)$ and $V_2(t)$ connect to a difference amplifier which consists of the resistors R_2, R_3, R_{N3}, R_{N4} and operational amplifier A_3. $V_2(t)$ subtracts $V_1(t)$ getting the output voltage of A_3

$$V_3(t) = \left(1 + \frac{R_{N3}}{R_3}\right)\left(\frac{\frac{R_{N4}}{R_2}}{1 + \frac{R_{N4}}{R_2}}\right)V_2(t) - \frac{R_{N3}}{R_3}V_1(t) \tag{7}$$

If the resistance of R_2, R_3, R_{N3} and R_{N4} are the same, the coefficients of $V_1(t)$ and $V_2(t)$ in (7) transfer as ‘1’

$$V_3(t) = V_2(t) - V_1(t) \tag{8}$$

According to (5) and (6), $V_3(t)$ can be rewritten as

$$V_3(t) = \left(\frac{R_{N1}}{R_m(t)} - \frac{R_{N2}}{R_1}\right)V_{I1} \tag{9}$$

V_{I1} is the input of the neuron while $V_3(t)$ is the output of the neuron. Hence, the synaptic weight is defined as

$$\omega(t) = \frac{R_{N1}}{R_m(t)} - \frac{R_{N2}}{R_1} \tag{10}$$

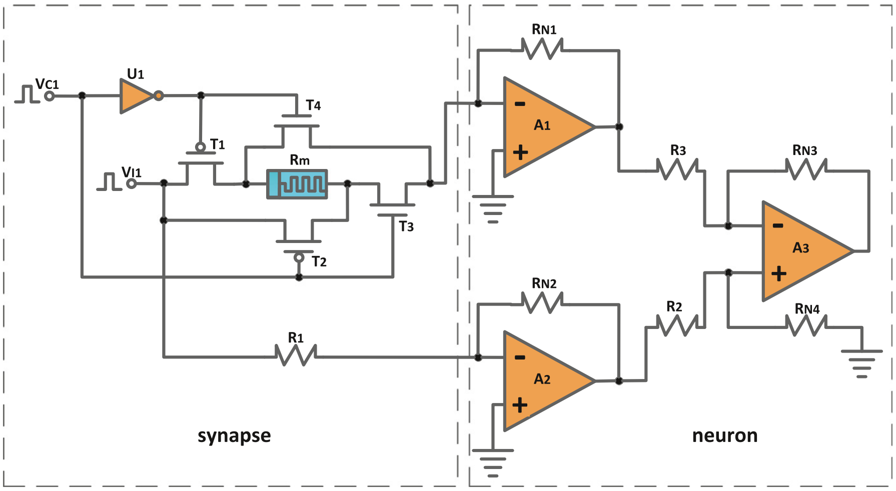


Fig. 1. The neuron circuit connecting one synaptic circuit.

Apparently, the negative weight, zero weight or positive weight is represented as

$$\omega(t)=\begin{cases}\text{negative weight,} & \text{if } \dfrac{R_{N1}}{R_m(t)}<\dfrac{R_{N2}}{R_1} \quad \text{(a)}\\ \text{zero weight,} & \text{if } \dfrac{R_{N1}}{R_m(t)}=\dfrac{R_{N2}}{R_1} \quad \text{(b)}\\ \text{positive weight,} & \text{if } \dfrac{R_{N1}}{R_m(t)}>\dfrac{R_{N2}}{R_1} \quad \text{(c)}\end{cases} \tag{11}$$

Because the resistance of R_1 and R_{N2} are fixed, (10) can be marked as

$$\omega(t)=\frac{R_{N1}}{R_m(t)}-a_0 \tag{12}$$

Taking the derivative of (4), we can get the relation between $R_m(t)$ and current $i_m(t)$

$$\frac{dR_m(t)}{dt}=-R_H\frac{\mu_v R_L}{D^2}i_m(t) \tag{13}$$

R_H, R_L, μ_v and D are constants in the memristor, so (13) is simplified as

$$\frac{dR_m(t)}{dt}=-k_0 i_m(t) \tag{14}$$

k_0 is given by

$$k_0=R_H\frac{\mu_v R_L}{D^2} \tag{15}$$

According to *Ohm's law*, $i_m(t)$ is given by

$$i_m(t)=\frac{V_m(t)}{R_m(t)} \tag{16}$$

where $V_m(t)$ denotes the voltage across the memristor. Then, a differential equation between $V_m(t)$ and $R_m(t)$ is obtained by combining (14) with (16)

$$\frac{dR_m(t)}{dt}=-k_0\frac{V_m(t)}{R_m(t)} \tag{17}$$

In the synaptic circuit, $V_m(t)$ is equal to the input signal V_{I1} whose voltage is a constant value at *logic* 1. Therefore, $R_m(t)$ is solved as

$$R_m(t)=\sqrt{C-2k_0V_{I1}t} \tag{18}$$

The initial memristance at $t=0$ is marked as $R_m(0)$. Then, $R_m(t)$ can be represented as

$$R_m(t)=\sqrt{(R_m(0))^2-2k_0V_{I1}t} \tag{19}$$

If the synaptic weight changes from w_1 to w_2, the memristance will vary accordingly

$$w_1=\frac{R_{N1}}{R_{md}}-a_0 \tag{20}$$

$$w_2=\frac{R_{N1}}{R_{mf}}-a_0 \tag{21}$$

The synaptic weight w_1 and w_2 are consistent with the memristance R_{md} and R_{mf}. The synaptic weight changes from w_1 to w_2, which indicates the memristance varies from R_{md} to R_{mf}. In the variation of the memristance, R_{md} is regarded as the initial memristance, and the memristance varies from R_{md} to R_{mf} during Δt

$$R_{mf}=\sqrt{(R_{md})^2-2k_0V_{I1}\Delta t} \tag{22}$$

Based on (20), (21) and (22), the lasting time of control signal-Δt is deduced as

$$\Delta t=\frac{R_{N1}^2(w_2+a_0)^2-R_{N1}^2(w_1+a_0)^2}{2k_0V_{I1}(w_1+a_0)^2(w_2+a_0)^2} \tag{23}$$

When the synaptic weight changes from w_1 to w_2, the duration of control signal-V_{C1} is calculated by (23). However, in [19–27,30–33] the relation between the synaptic weight change and the lasting time of control signals is not given, which is disadvantageous for the training of the networks.

2.2. Neuron circuit connecting two synaptic circuits

The circuit implementation of the neuron circuit which connects two synaptic circuits is shown in Fig. 2. When $V_{I1}=V_{I2}$=0, the memristance of R_{m1} and R_{m2} keep unchanged, and the output voltages of A_1 and A_2 are 0V. When V_{I1}=0, V_{I2}=1, the voltage across the memristor R_{m1} is 0V, so no current flows through R_{m1}. The voltage on R_1 is also 0V. The synaptic circuit which includes R_{m1} makes no contribution to the output voltages of A_1 and A_2. In this condition, only the synaptic circuit which contains R_{m2} influences the output voltages of A_1 and A_2. When $V_{I1}=V_{I2}$=1, the effects of the two synaptic circuits are added

$$V_1(t)=-\left(\frac{V_{I1}R_{N1}}{R_{m1}(t)}+\frac{V_{I2}R_{N1}}{R_{m2}(t)}\right) \tag{24}$$

$$V_2(t)=-\left(\frac{V_{I1}R_{N2}}{R_1}+\frac{V_{I2}R_{N2}}{R_2}\right) \tag{25}$$

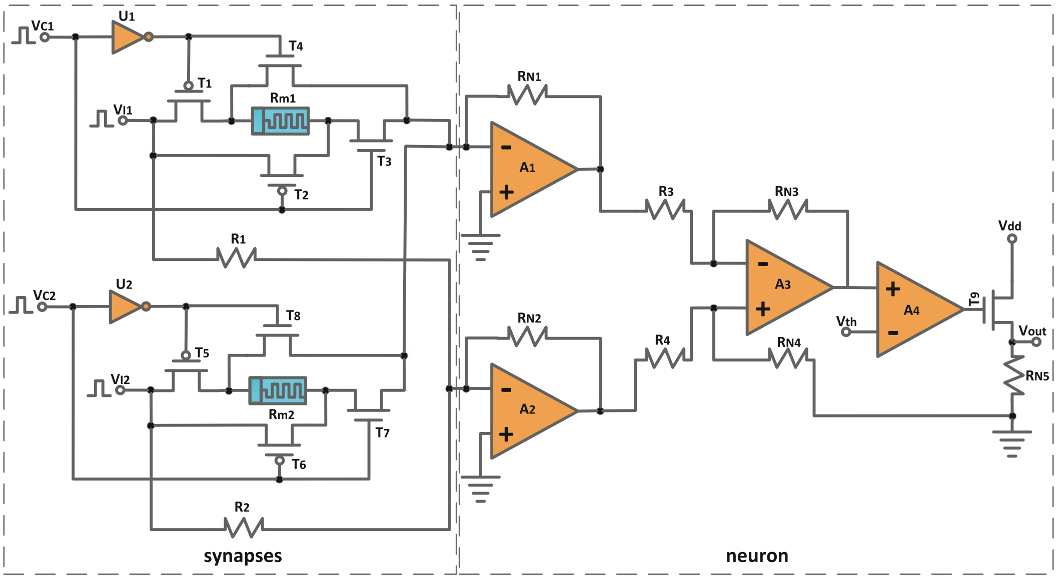


Fig. 2. The circuit implementation of the neuron circuit connecting two synaptic circuits.

$V_3(t)$ is given by

$$V_3(t) = \left(\frac{R_{N1}}{R_{m1}(t)} - \frac{R_{N2}}{R_1}\right)V_{I1} + \left(\frac{R_{N1}}{R_{m2}(t)} - \frac{R_{N2}}{R_2}\right)V_{I2} \tag{26}$$

Similarly, the relation between inputs and output of the neuron circuit connecting n synaptic circuits can be represented as

$$V_3(t) = \sum_{i=1}^{n}\left(\frac{R_{N1}}{R_{mi}(t)} - \frac{R_{N2}}{R_i}\right)V_{Ii} \tag{27}$$

The synaptic weight is defined as

$$\omega_i(t) = \frac{R_{N1}}{R_{mi}(t)} - \frac{R_{N2}}{R_i} \tag{28}$$

The memristance varies in the certain range, and if we change the resistance of R_{N1}, R_{N2} and R_i, the variation scope of the synaptic weight will change accordingly. However, variation range of the synaptic weight in [19,20] is restricted in (-1,1), which limits the application of the memristor-based neural network circuits.

In Fig. 2, A_4 is a comparator whose threshold voltage is V_{th}. The output voltage of A_4 is transferred to digital logic level by the transistor T_9 and the resistor R_{N5}. Thus, the activation function of the neuron circuit is given by

$$sign(V_3(t) - V_{th}) = \begin{cases} 1, & \text{if } V_3(t) - V_{th} \geq 0 \quad \text{(a)} \\ 0, & \text{if } V_3(t) - V_{th} < 0 \quad \text{(b)} \end{cases} \tag{29}$$

The PSPICE simulation of the neuron circuit which connects two synaptic circuits is shown in Fig. 3. In the simulation circuit, the resistance of R_1, R_2, R_{N1} and R_{N2} are set as $50K\Omega$, the resistance of R_3, R_4, R_{N3} and R_{N4} are arranged as $500K\Omega$, the initial memristance of R_{m1} and R_{m2} are set as $60K\Omega$, V_{th} is set as 1V. For *HP* memristor, R_H, R_L, μ_v and D are arranged as $100K\Omega$, $1K\Omega$, 10nm and $10^{-14}m^2s^{-1}V^{-1}$, respectively [36]. In Fig. 3, R_{m1} and R_{m2} indicate the memristance of the two memristors. V_{A1}, V_{A2} and V_{A3} represent the output voltages of A_1, A_2 and A_3, respectively.

In 0–20 ms, both of the input signals are *logic* 0, $V_{I1} = V_{I2}$=0. Thus, V_{A1}, V_{A2} and V_{A3} are 0V, and the memristance of R_{m1} and R_{m2} remain invariant.

In 20–60 ms, V_{I1} is *logic* 1 while V_{I2} *logic* 0. Hence, the synaptic circuit which contains R_{m2} does not influence the output of the neuron circuit. In this period, the control signal V_{c1} lasts the equal time at *logic* 0 and *logic* 1, so memristance of R_{m1} presents symmetrical variation. The change of V_{A1} is symmetrical thereby.

At 20 ms, V_{I1} changes from *logic* 0 to *logic* 1 instantly. Thus, V_{A1}, V_{A2} and V_{A3} generate instantaneous jumps at 20 ms. The values of V_{A1}, V_{A2} and V_{A2} at 20 ms are given by

$$V_2(20) = -\frac{V_{I1}R_{N2}}{R_1} = -5V \tag{30}$$

$$V_1(20) = -\frac{V_{I1}R_{N1}}{R_{m1}(20)} = -4.17V \tag{31}$$

$$V_3(20) = V_2(20) - V_1(20) = -0.83V \tag{32}$$

The memristance of R_{m1} reduces from $60\,K\Omega$ during 20–40 ms, so V_{A3} rises in this period. The memristance of R_{m1} and the value of V_{A3} at 40 ms can be inferred as

$$R_{m1}(40) = \sqrt{(R_{m1}(20))^2 - 0.04k_0V_{I1}} = 40K\Omega \tag{33}$$

$$V_3(40) = \left(\frac{R_{N1}}{R_{m1}(40)} - \frac{R_{N2}}{R_1}\right)V_{I1} = 1.25V \tag{34}$$

The memristance of R_{m1} increases from $R_{m1}(40)$ to $60\,K\Omega$ in 40–60 ms, and V_{A3} reduces in this period. In 20–60 ms, the value of V_{A3} is greater than the threshold voltage 1V for a period, making V_{out} have a pulse during 20–60 ms.

In 60–100 ms, both of the input signals are *logic* 1, $V_{I1} = V_{I2}$=1. In this period, control signals V_{C1} and V_{C2} have the same change and last equal time at *logic* 0 and *logic* 1. Therefore, the change of R_{m1}, R_{m2}, V_{A1}, and V_{A3} are symmetrical during this period. The effects of the two synaptic circuits are added by the neuron circuit. Thus, the variations of V_{A1} and V_{A3} in this period are faster than that in 20-60ms.

In 60-100ms, V_{out} is the response of the neuron circuit to the two input signals. According to the logic levels of V_{C1} and V_{C2}, the memristance of R_{m1} and R_{m2} decrease and increase in 60–80 ms and 80–100 ms, which indicate adjustments of the synaptic weights. As a consequence, the proposed neuron circuit can respond to input signals and adjust synaptic weights at the same time, because its input terminals and its control signal terminals are separated. However, in [19–29], the response to input signals and the synaptic weight adjustment of the neuron circuit are

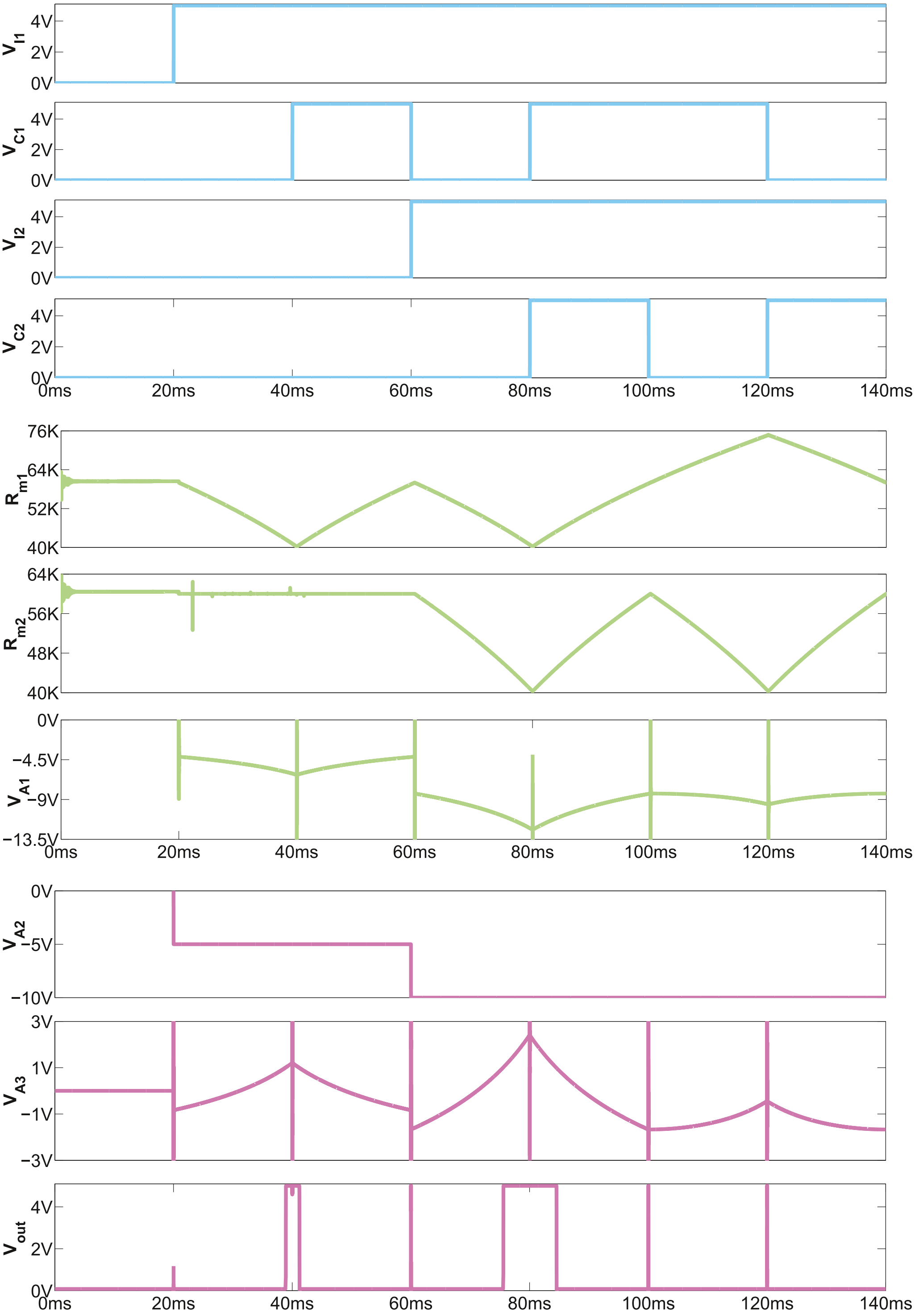


Fig. 3. The simulation of the neuron circuit which connects two synaptic circuits.

arranged at the same terminal, so the response and the adjustment are accomplished in several steps.

At 60ms, V_{I2} changes to *logic* 1 instantly, and V_{I1} remains at *logic* 1. Therefore, V_{A1}, V_{A2} and V_{A3} hop to -8.34V, -10V and -1.66V, respectively

$$V_1(60) = -\left(\frac{V_{I1}R_{N1}}{R_{m1}(60)} + \frac{V_{I2}R_{N1}}{R_{m2}(60)}\right) = -8.34V \tag{35}$$

$$V_2(60) = -\left(\frac{V_{I1}R_{N2}}{R_1} + \frac{V_{I2}R_{N2}}{R_2}\right) = -10V \tag{36}$$

$$V_3(60) = V_2(60) - V_1(60) = -1.66V \tag{37}$$

In 60–100 ms, the memristance of R_{m1} and R_{m2} vary between 40 $K\Omega$ and 60 $K\Omega$. V_{A3} changes between -1.66 V and 2.5 V. Because V_{A3} exceeds the threshold voltage 1 V for a period, V_{out} includes a pulse.

In 100–140 ms, the two input signals remain at *logic* 1, while V_{C1} and V_{C2} have opposite change in this period. At 100 ms, the memristance of R_{m1} and R_{m2} are 60 $K\Omega$. Under the control of V_{C1}

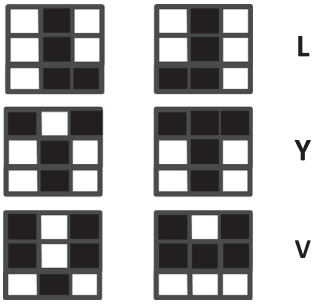


Fig. 4. The pictures of the characters ‘L’, ‘Y’, ‘V’.

and V_{C2}, the memristance of R_{m1} increases while the memristance of R_{m2} decreases during 100–120 ms. The memristance of R_{m1} and R_{m2} at 120 ms can be represented as

$$R_{m1}(120) = \sqrt{(R_{m1}(100))^2 + 0.04k_0V_{I1}} = 74.83K\Omega \tag{38}$$

$$R_{m2}(120) = \sqrt{(R_{m2}(100))^2 - 0.04k_0V_{I1}} = 40K\Omega \tag{39}$$

V_{A3} at 120 ms is deduced as

$$\begin{aligned} V_3(120) &= \left(\frac{R_{N1}}{R_{m1}(120)} - \frac{R_{N2}}{R_1}\right)V_{I1} + \left(\frac{R_{N1}}{R_{m2}(120)} - \frac{R_{N2}}{R_2}\right)V_{I2} \\ &= -0.41V \end{aligned} \tag{40}$$

Then, the memristance of R_{m1} and R_{m2} return to 60 $K\Omega$ by the control of V_{C1} and V_{C2} during 120–140 ms. The summed effect of the two synaptic circuits makes the value of V_{A3} rise in 100–120 ms and reduce in 120–140 ms. Since V_{A3} is less than the threshold voltage 1 V, V_{out} does not have pulse during 100–140 ms.

As analyzed above, the calculation values of the circuit parameters fit with the simulation curves approximately, which demonstrates that the circuit implementation and the analysis of the neuron circuit are effective. However, in [19–33], the circuit parameter changes of the neuron circuits are unknown, so it is disadvantageous for the training or extension of the memristor-based neural network circuits which consist of these neuron circuits.

3. Character recognition network

3.1. Circuit implementation of the character recognition network

Based on the proposed neuron circuit, a memristor-based neural network circuit is devised to realize character recognition. Because the neuron circuit is compatible with CMOS logic level, a microcontroller is utilized to control the operation of the network. The characters to be recognized include standard pictures and the pictures with noise, which are shown in Fig. 4. After training, when a picture of the characters is input to the recognition network, it is recognized as ‘L’, ‘Y’ or ‘V’.

The schematic diagram of the character recognition network is shown in Fig. 5. As shown in the figure, before a picture is applied to the character recognition network, it is mapped to one dimension vector to match the circuit structure. The mapped nine input signals contact with the three neurons *NEU*1, *NEU*2 and *NEU*3 in full-connection form. Each neuron circuit has nine synaptic circuits to correlate the nine inputs.

When one of the pictures is input to the recognition network, the mapped input signals make dot product with the corresponding synaptic weights to get summed voltages. The summed voltages not only send to the activation function obtaining the outputs of network but also return to the microcontroller for synaptic weight adjustment. For a neuron circuit, the summed voltage can be represented as

$$f = \boldsymbol{wp} \tag{41}$$

where $\boldsymbol{w}$ denotes the synaptic weight vector of the neuron, $\boldsymbol{p}$ is the vector composed of the input signals.

Widrow-Hoff algorithm is adopted to train the character recognition network [37].

Widrow-Hoff algorithm is a basic supervised learning algorithm to train SNN, which is also called as adaline algorithm or δ rule [21,30,37,38]. The algorithm utilizes mean square error (MSE) to serve as loss function

$$\boldsymbol{e^2} \triangleq \sum \|\boldsymbol{t} - \boldsymbol{f}\|^2 \tag{42}$$

where $\boldsymbol{e}$ is the error vector, $\boldsymbol{t}$ is the target vector, $\boldsymbol{f}$ is the vector constituted by summed voltages.

Through minimizing MSE of the network, the network can achieve aimed function. In Widrow-Hoff algorithm, the minimization of MSE is along with the direction of gradient. The gradient of one neuron can be represented as

$$\frac{\partial e^2}{\partial w_{i,j}} = 2e\frac{\partial e}{\partial w_{i,j}} \tag{43}$$

Combining (41) and (42), the derivative between error and synaptic weight is given by

$$\frac{\partial e}{\partial w_{i,j}} = -p_j \tag{44}$$

As a consequence, the weight iteration rule of Widrow-Hoff algorithm is given by

$$\boldsymbol{f}(k) = \boldsymbol{W}(k)\boldsymbol{p}(k) \tag{45}$$

$$\boldsymbol{e}(k) = \boldsymbol{t}(k) - \boldsymbol{f}(k) \tag{46}$$

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) + 2\alpha\boldsymbol{e}(k)\boldsymbol{p}^T(k) \tag{47}$$

where $\boldsymbol{W}(k)$ represents the current weight matrix while $\boldsymbol{W}(k+1)$ indicates the weight matrix in the next step, α is the learning rate. The increment or decrement of synaptic weight is determined by the error and input signal.

According to the relation between voltage and current, the recognition network gets $\boldsymbol{f}(k)$ immediately. The remainder procedures of Widrow-Hoff algorithm are operated on the microcontroller to accomplish an iteration.

The black pixels of one picture are mapped as *logic* 1 while the white pixels are mapped as *logic* 0. For example, the standard picture of character ‘L’ is mapped as $\boldsymbol{p}_1 = \begin{bmatrix} 0 & 0 & 0 & 5 & 5 & 5 & 0 & 0 & 5 \end{bmatrix}^T$. The threshold voltages of the neuron circuits are set as 2V. The target vectors of ‘L’, ‘Y’ and ‘V’ are arranged as $\boldsymbol{t}_1 = \begin{bmatrix} 3 & 1 & 1 \end{bmatrix}^T$, $\boldsymbol{t}_2 = \begin{bmatrix} 1 & 3 & 1 \end{bmatrix}^T$, and $\boldsymbol{t}_3 = \begin{bmatrix} 1 & 1 & 3 \end{bmatrix}^T$. After the training, when the pictures of character ‘L’ are applied to the recognition network, *NEU*1 outputs *logic* 1 while *NEU*2 and *NEU*3 output *logic* 0.

After $\boldsymbol{f}(k)$ is obtained, the microcontroller computes the synaptic weight variations and adjusts synaptic weights synchronously. The specific procedures are given as follows:

(1) The microcontroller converts $\boldsymbol{f}(k)$ as digital signals by the A/D converter, and then the error vector $\boldsymbol{e}(k)$ is got.
(2) The microcontroller utilizes the input signal vector $\boldsymbol{p}(k)$ and $\boldsymbol{e}(k)$ to calculate the weight matrix in the next step.

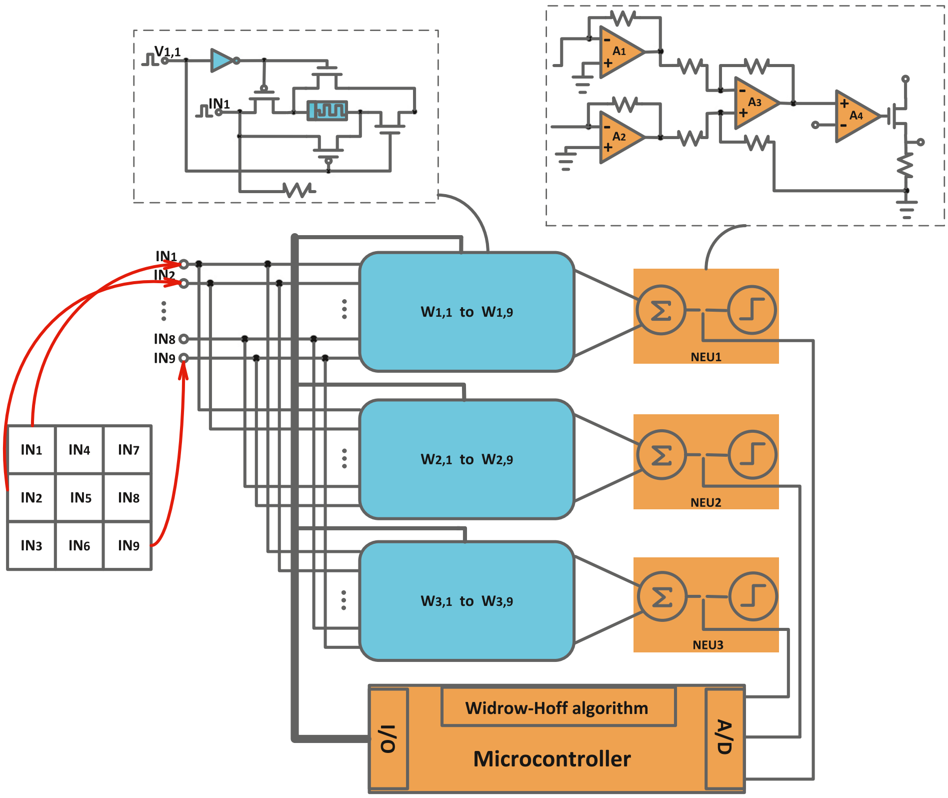


Fig. 5. The schematic diagram of the character recognition network.

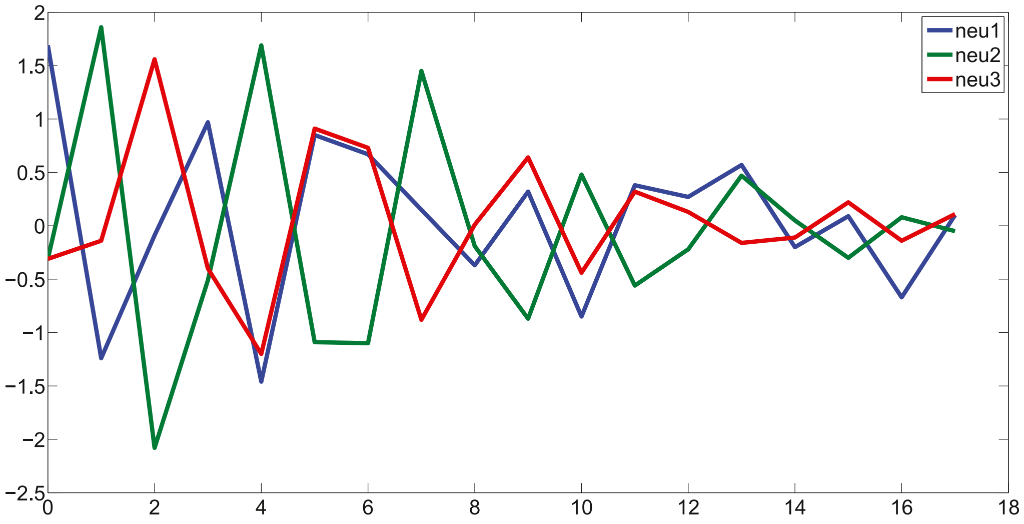


Fig. 6. The errors between the targets and summed voltages.

(3) Based on the current weights and the weights in the next step, combining the relation presented in (23), the microcontroller computes the duration of the control signals $\boldsymbol{T}_c(k)$.

(4) From the beginning of (1) to the end of (3), this period is marked as T_w. $\boldsymbol{T}_c(k)$ adds or subtracts T_w to get the eventual duration of the control signals $\boldsymbol{T}_{ce}(k)$. Then, I/O ports of the microcontroller output the corresponding digital signals lasting $\boldsymbol{T}_{ce}(k)$ to adjust weights synchronously.

In the recognition network, all the initial synaptic weights are set as 0.05. Then, the pictures to be recognized are applied to train the character recognition network. After 17 times iterations, all the pictures are recognized correctly. Because $\boldsymbol{f}(k)$ is got directly, the complexity of the algorithm is O(17). The recognition network is simulated on PSPICE to prove the performance. The errors between the targets and summed voltages reduce along with the iterations. The errors of the three neuron circuits are shown in Fig. 6.

The synaptic circuits in the character recognition network can be adjusted synchronously. In an iteration, the synaptic circuits required to change are adjusted at the same time, so the iteration is accomplished with one adjustment. However, for the majority of the existing memristor-based neural network circuits, the synaptic circuits cannot be adjusted synchronously in an iteration, the iteration is realized by several adjustments and several steps. This slows down training speed.

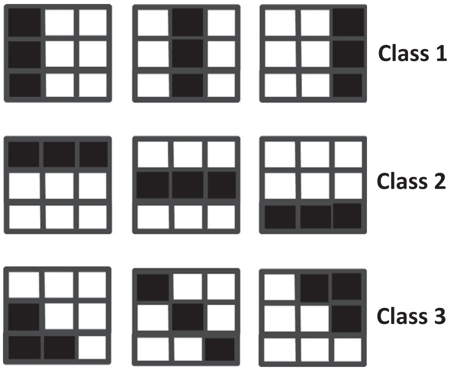


Fig. 7. The pictures of three classes.

4. Pattern classification network

4.1. Circuit implementation of the pattern classification network

SNN is capable of solving linear separable problems. For linear inseparable problems, they are settled by MNN. Based on the proposed neuron circuit, a memristor-based MNN circuit is constructed to achieve a linear inseparable classification. The pictures to be classified are shown in Fig. 7. Because the black pixels of the three pictures in one class occupy different positions of nine pixels and the patterns in the three classes are also different, the classification is a linear inseparable problem [37].

The structure of the pattern classification network is presented in Fig. 8. As shown in the figure, the classification network is a part-connection network. In the first layer, the black pixels of a picture are mapped to input signals to contact with a neuron. $N_1 - N_3$, $N_4 - N_6$ and $N_7 - N_9$ are related to the pictures in *Class* 1, *Class* 2 and *Class* 3, respectively. For the second layer, only a part of inputs connect to the corresponding neuron. If OT_1 is *logic* 1 and OT_2, OT_3 are *logic* 0, it implies that the input picture belongs to *Class* 1.

In the first layer, the mapped input signals are the black pixels of a picture. Because the pictures to be classified have black pixels at the same position, the synaptic weights must satisfy a rule to realize correct classification. The black pixels are mapped to *logic* 1 whose voltage is 5V and the threshold voltage of the neuron circuits is 2V. Therefore, the rule is given by

$$rul\begin{cases}5\times(w^1_{a,i}+w^1_{a,j}+w^1_{a,m})>2 & \text{(a)}\\ 5\times(w^1_{a,i}+w^1_{a,i})<2 & \text{(b)}\\ 5\times(w^1_{a,i})<2 & \text{(c)}\end{cases} \tag{48}$$

If the synaptic weights in the first layer are the same after training, the range of the synaptic weights is given by

$$0.133 < w^1 < 0.2 \tag{49}$$

Here, the synaptic weights is selected as 0.16. Then, we can calculate the duration of control signals according to initial synaptic weights and (23).

When a picture is applied to the classification network, only one neuron in the first layer outputs *logic* 1. Accordingly, the related neuron in second layer outputs *logic* 1. Hence, the synaptic weights in the second layer should satisfy a rule

$$5\times w^2_{i,j}>2 \tag{50}$$

Similarly, if the synaptic weights in the second layer are the same after training, the synaptic weights is selected as 0.45. Further, we can get the lasting time of the control signals.

Table 1
The relation between the selected voltage and C_r, S_w.

C_r	S_w	*voltage*
0	0	V_{th}
0	1	V_{th}
1	0	V_{ff}
1	1	V_{ss}

4.2. Synchronous weight adjustment of the pattern classification network

In pattern classification network, only the input signal of a synaptic circuit is *logic* 1, the corresponding synaptic weight can be adjusted to increase or decrease. The outputs of the first layer are the input signals of synaptic circuits of the second layer. In order to achieve synchronous weight adjustment, the input signals of synaptic circuits in the second layer should be able to set as *logic* 1. Hence, the neuron circuits in the first layer are modified by connecting *voltage selection circuits* which control the outputs of the first layer. The schematic diagram of the pattern classification network is shown in Fig. 9, and the modified neuron circuit is presented at the left bottom.

The *voltage selection circuit* has three input voltages. V_{th} is the threshold voltage of the neuron circuit. V_{ff} is the positive power supply voltage of the comparator while V_{ss} is the negative power supply voltage. C_r denotes that the network operates in response to input signals or weight adjustment. S_w is the control signal to arrange the neuron circuit in the first layer as *logic* 1 or *logic* 0. The truth table between the selected voltage and C_r, S_w is shown in Table 1.

The synchronous weight adjustment of the pattern classification network is given as follows. First, C_r is arranged as *logic* 0, which indicates the network responds to input signals. The voltage of *voltage selection circuit* is selected as V_{th}, and the mapped input signals are applied to the network to obtain the outputs. Then, C_r is set as *logic* 1, which denotes the network operates in weight adjustment. The microcontroller computes the lasting time of control signals. When S_w is arranged as *logic* 1, the voltage of *voltage selection circuit* is selected as V_{ss}. The output of the corresponding neuron circuit is set to *logic* 1 thereby. Hence, the related synaptic circuit in the second layer can adjust weight. When S_w is arranged as *logic* 0, the corresponding neuron circuit outputs *logic* 0. Next, the microcontroller outputs control signals of the synaptic circuits to adjust weights synchronously. The synaptic circuits required to adjust in the pattern classification network can adjust weights at the same time.

The comparisons of the different memristor-based neural network circuits are presented in Table 2. For the memristor-based neural network circuits in [28,29], in an iteration, only one synaptic circuit can adjust weight at a time. If n synaptic circuits need to change weights in an iteration of SNN, the iteration is accomplished with n adjustments. Further, the adjustment times of an iteration increase along with the layers of MNN. It assumes that the average consuming time of one synaptic circuits adjustment is T_m. For an iteration of SNN, the consuming time is $n \times T_m$. However, because the proposed memristor-based neural network circuits can adjust synaptic weights at the same time in an iteration, the consuming time of the iteration is T_m. No matter in the proposed SNN or MNN, the synaptic weights are adjusted synchronously, which improves training speed.

In [25–27], the synaptic circuits represented by the cross-bar array can only adjust one row at a time, and this process is accomplished by several steps. Therefore, the training speed is slowed down.

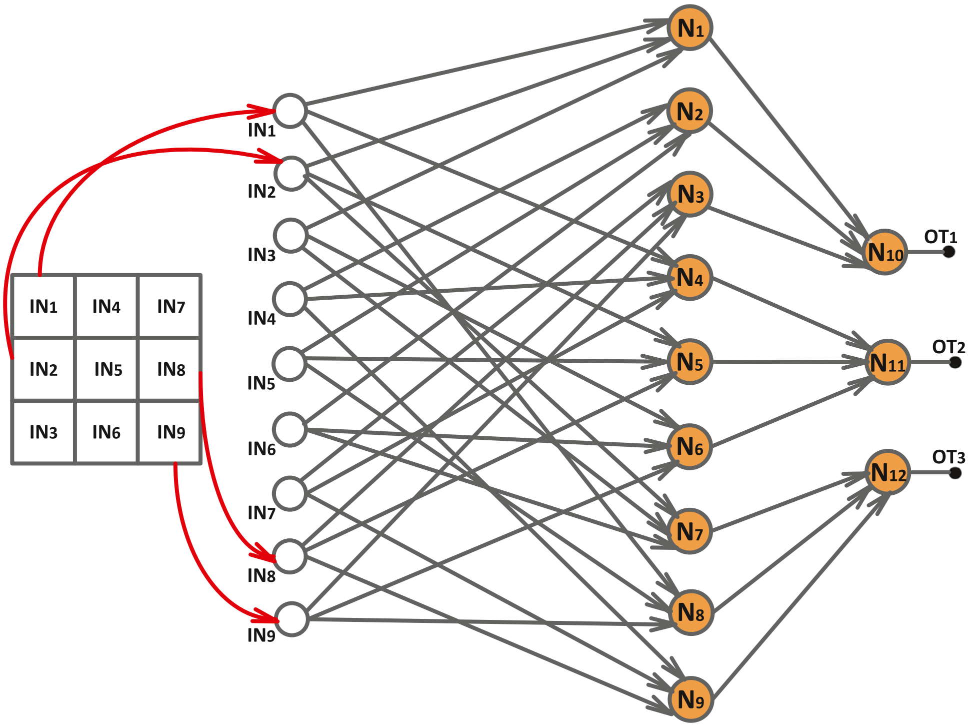


Fig. 8. The structure of the pattern classification network.

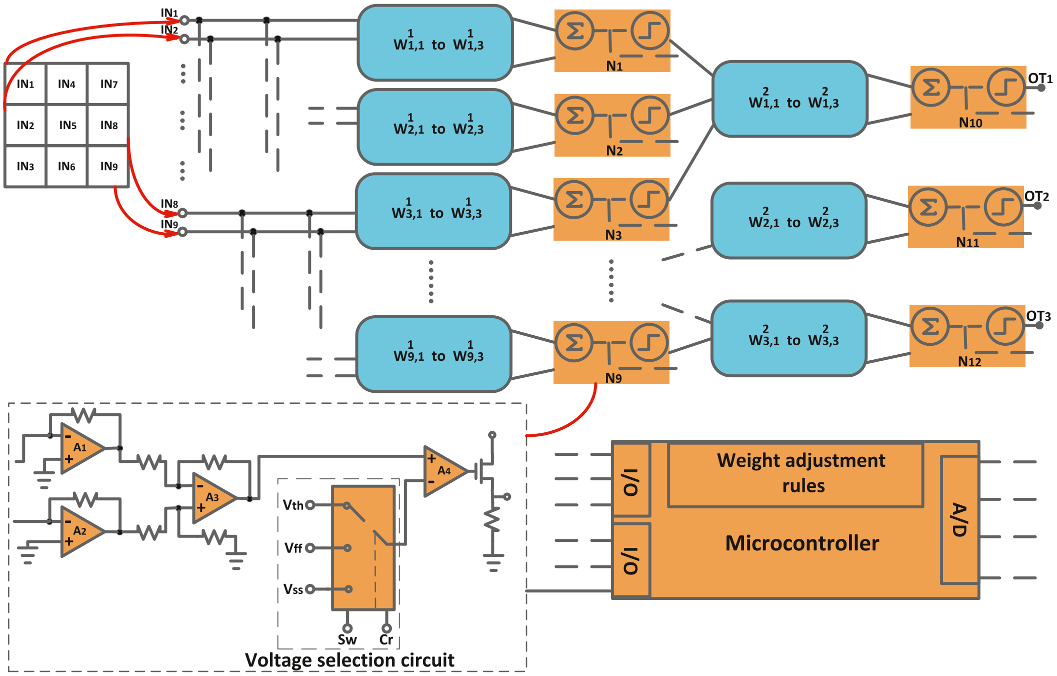


Fig. 9. The schematic diagram of the pattern classification network.

Table 2
The comparisons of the different memristor-based neural network circuits.

	[28,29]	[25–27]	[30–33]	[19],[20]	this paper
synaptic weight	positive, zero and negative	positive, zero and negative	positive only	positive, zero and negative	positive, zero and negative
synaptic weight adjustment	one at a time	one row at a time	one layer at a time	–	synchronous
range of synaptic weight	changeable	changeable	unchangeable positive range	(−1, 1)	changeable
relation between control signal and synaptic weight variation	given	unknown	unknown	unknown	given

In [30,32,33], when the synaptic weight changes to negative or zero weight, the memristance of the synaptic circuit is mapped to negative or zero value which cannot be realized by physical memristor. In [19,20], restricted by the circuit structure of synaptic circuit, the variation range of synaptic weight is limited in (-1,1). This restricts the extension and application of the memristor-based neural network circuits. Moreover, in majority of the existing memristor-based neural network circuits, the relation between duration of control signals and synaptic weight change is unknown, which is disadvantageous for the training of the networks.

However, the proposed neuron circuit in this paper can not only change the range of synaptic weight by arranging the corresponding resistance, but also present the variation of synaptic weight along with the lasting time of control signal. The synaptic circuit represents positive, zero or negative weight following the memristance change.

5. Conclusions

Memristor-based neural network circuits are considered as a promising candidate for hardware implementation of computing acceleration. This paper proposes a neuron circuit whose synaptic circuits are controlled by digital logic levels. According to the circuit structure of the neuron circuit, the relation between the synaptic weight change and the duration of the control signal is inferred. Based on the neuron circuit, a memristor-based SNN circuit is designed to achieve character recognition. Widrow-Hoff algorithm is adopted to train the SNN. Benefitting from the circuit structure, the SNN can realize synchronous weight adjustment in an iteration. Then, a memristor-based MNN circuit is constructed to achieve a linear inseparable classification. Through modifying the circuit structure of the neuron circuits in the first layer, the MNN can achieve synchronous weight adjustment in an iteration which improves training speed. The proposed circuits are emulated on PSPICE to prove the effectiveness.

Declarations of interest

None.

Acknowledgements

The work was supported by the Natural Science Foundation of China under Grants 61673188, 61761130081 and 61821003, the National Key Research and Development Program of China under Grant 2016YFB0800402, the Foundation for Innovative Research Groups of Hubei Province of China under Grant 2017CFA005, the Fundamental Research Funds for the Central Universities of HUST under Grant 2018KFYXKJC051 and Newton Advanced Fellowship Award of the Royal Society under Grants NA160545.

References

[1] X. Hu, G. Feng, S. Duan, L. Liu, A memristor-based multilayer cellular neural network with applications to image processing, IEEE Trans. Neural Netw. Learn. Syst. 28 (8) (2017) 1889–1901.

[2] S. Hu, Y. Liu, Z. Liu, T. Chen, J. Wang, Q. Yu, et al., Associative memory realized by a reconfigurable memristor-based hopfield neural network, Nat. Commun. 6 (7522) (2015) 1–8.

[3] J. Chen, J.H. Park, J. Cao, D. Zhang, Quantized H∞ filtering for switched linear parameter-varying systems with sojourn probabilities and unreliable communication channels, Inf. Sci. 466 (2018) 289–302.

[4] J. Chen, C.K. Anh, H.R. Karimi, J. Cao, W. Qi, An event-based asynchronous approach to markov jump systems with hidden mode detections and missing measurements, IEEE Trans. Syst. Man Cybern. Syst. (2018), doi:10.1109/TSMC.2018.2866906.

[5] L.O. Chua, Memristor-the missing circuit element, IEEE Trans. Circuit Theory 18 (5) (1971) 507–519.

[6] D.B. Strukov, G.S. Snider, G.R. Stewart, R.S. Williams, The missing memristor found, Nature 453 (7191) (2008) 80–83.

[7] X. Shi, S. Duan, L. Wang, T. Huang, C. Li, A novel memristor-based electronic synapse-based hermite chaotic neural network with application in cryptography, Neurocomputing 166 (2015) 487–495.

[8] L. Yang, Z. Zeng, Y. Huang, S. Wen, Memristor-based circuit implementations of recognition network and recall network with forgetting stages, IEEE Trans. Cognit. Dev. Syst. 10 (4) (2018) 1133–1142.

[9] L. Guckert, E.E. Swartzlander, MAD gates memristor logic design using driver circuitry, IEEE Trans. Circuits Syst. II Exp. Briefs 64 (2) (2017) 171–175.

[10] M.A. Zidan, H. Omran, A.G. Radwan, K.N. Salama, Memristor-based reactance–less oscillator, Electron. Lett. 47 (22) (2015) 1220–1221.

[11] P. Zhang, C. Li, T. Huang, L. Chen, Y. Chen, Forgetting memristor based neuromorphic system for pattern training and recognition, Neurocomputing 222 (2017) 47–53.

[12] M. Hu, Y. Chen, J.J. Yang, Y. Wang, H.H. Li, A compact memristor-based dynamic synapse for spiking neural networks, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 36 (8) (2017), 1353–1366.

[13] M. Elshamy, H. Mostafa, Y.H. Ghallab, M.S. Said, A novel nondestructive read/write circuit for memristor-based memory arrays, IEEE Trans. Very Large Scale Integr. Syst. 23 (11) (2015) 2648–2656.

[14] S.-J. Lee, S.-J. Kim, K. Cho, et al., Complementary resistive switch-based smart sensor search engine, IEEE Sens. J. 14 (5) (2014) 1639–1646.

[15] S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (4) (2010) 1297–1301.

[16] J.A. Starzyk, Basawaraj, Memristor crossbar architecture for synchronous neural networks, IEEE Trans. Circuits Syst. I Regul. Pap. 61 (8) (2014) 2390–2401.

[17] L. Chen, C. Li, T. Huang, Y. Chen, X. Wang, Memristor crossbar-based unsupervised image learning, Neural Comput. Appl. 25 (2) (2014) 393–400.

[18] L. Yang, Z. Zeng, S. Wen, A full-function pavlov associative memory implementation with memristance changing circuit, Neurocomputing 272 (2018) 513–519.

[19] S.P. Adhikari, C. Yang, H. Kim, L.O. Chua, Memristor bridge synapse-based neural network and its learning, IEEE Trans. Neural Netw. Learn. Syst. 23 (9) (2012) 1426–1435.

[20] 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.

[21] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K.K. Likharev, D.B. Strukov, Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521 (7550) (2015) 61–64.

[22] F. Alibart, E. Zamanidoost, D.B. Strukov, Pattern classification by memristor-based crossbar circuits using ex situ and in situ training, Nat. Commun. 4 (2013) 2072–2079.

[23] M. Hu, H. Li, Y. Chen, Q. Wu, G.S. Rose, R.W. Linderman, Memristor crossbar-based neuromorphic computing system: a case study, IEEE Trans. Neural Netw. Learn. Syst. 25 (10) (2014) 1864–1878.

[24] I. Kataeva, F. Merrikh-Bayat, E. Zamanidoost, D. Strukov, Efficient training algorithms for neural networks based on memristive crossbar circuits, in: Proceedings of the International Joint Conference on Neural Networks, 2015, pp. 1–8.

[25] D. Chabi, Z. Wang, C. Bennett, J.O. Klein, W. Zhao, Ultrahigh density memristor neural crossbar for on-chip supervised learning, IEEE Trans. Nanotechnol. 14 (6) (2015) 954–962.

[26] R. Hasan, T.M. Taha, Enabling back propagation training of memristor crossbar neuromorphic processors, in: Proceedings of the International Joint Conference on Neural Networks, 2014, pp. 21–28.

[27] R. Hasan, T.M. Taha, C. Yakopcic, A fast training method for memristor crossbar based multi-layer neural networks, Analog Integr. Circuits Sig. Process. 93 (3) (2017) 443–454.

[28] Y. Zhang, Y. Li, X. Wang, E.G. Friedman, Synaptic characteristics of Ag/AgInSbTe/Ta-based memristor for pattern recognition applications, IEEE Trans. Electron Devices 64 (4) (2017) 1806–1811.

[29] 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.

[30] 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.

[31] S. Xiao, X. Xie, S. Wen, Z. Zeng, T. Huang, J. Jiang, GST-memristor-based online learning neural networks, Neurocomputing 272 (2018) 677–682.

[32] S. Wen, S. Xiao, Z. Yan, Z. Zeng, T. Huang, Adjusting learning rate of memristor-based multilayer neural networks via fuzzy method, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 38 (6) (2019) 1084–1094.

[33] S. Wen, R. Hu, Y. Yang, T. Huang, Z. Zeng, Y. Song, Memristorbased echo state network with online least mean square, IEEE Tran. Syst. Man, Cybern. Syst. (2018), doi:10.1109/TSMC.2018.2825021.

[34] H. Kang, Fundamentals of Electronic Technology, Higher Education Press, Beijing, 2006.

[35] D.A. Neamen, Micoelectronics: Circuit Analysis and Design, McGraw-Hill, Boston, 2007.

[36] Z. Biolek, D. Biolek, V. Biolkova, Spice model of memristor with nonlinear dopant drift, Radioengineering 18 (2) (2009) 210–214.

[37] M.T. Hagan, H.B. Demuth, M. Beale, Neural Network Design, PWS Publishing, Boston, MA, 1996.

[38] B. Widrow, M.E. Hoff, Adaptive switching circuits, Stanford Electron. Labs, Stanford University, Stanford, CA, USA, Technical Report, 1960.

Le Yang received the M.S. degree in the School of Engineering, Huazhong Agricultural University, Wuhan, China, in 2014. He is currently working toward the D.S. degree in the School of Automation, Huazhong University of Science and Technology, Wuhan, China. His main interest is memristor-based artificial neural network.

Zhigang Zeng received the Ph.D. degree in systems analysis and integration from Huazhong University of Science and Technology, Wuhan, China, in 2003. He is currently a Professor with the School of Automation, Huazhong University of Science and Technology, Wuhan, China, and also with the Key Laboratory of Image Processing and Intelligent Control of the Education Ministry of China, Wuhan, China. He has been an Associate Editor of the IEEE Transactions on Neural Networks (2010–2011), IEEE Transactions on Cybernetics (since 2014), IEEE Transactions on Fuzzy Systems (since 2016), and a member of the Editorial Board of Neural Networks (since 2012), Cognitive Computation (since 2010), Applied Soft Computing (since 2013). He has published over 100 international journal papers. His current research interests include theory of functional differential equations and differential equations with discontinuous right-hand sides, and their applications to dynamics of neural networks, memristive systems, and control systems.

Xinming Shi received the B.S. degree from the College of Electrical Engineering, Wuhan University of Technology, Wuhan, China, in 2016. She is currently working toward the M.S. degree at the School of Automation, Huazhong University of Science and Technology, Wuhan, China. Her research focuses on memristor-based artificial neural network circuits.