


Article

A Novel Memristive Neural Network Circuit and Its Application in Character Recognition

Xinrui Zhang ¹, Xiaoyuan Wang ^{1,*}, Zhenyu Ge ¹, Zhilong Li ¹, Mingyang Wu ¹ and Shekharsuman Borah ²¹ School of Electronics and Information, Hangzhou Dianzi University, Hangzhou 310018, China² Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Guwahati 781015, India

* Correspondence: youyuan-0213@163.com

Abstract: The memristor-based neural network configuration is a promising approach to realizing artificial neural networks (ANNs) at the hardware level. The memristors can effectively simulate the strength of synaptic connections between neurons in neural networks due to their diverse significant characteristics such as nonvolatility, nanoscale dimensions, and variable conductance. This work presents a new synaptic circuit based on memristors and Complementary Metal Oxide Semiconductor (CMOS), which can realize the adjustment of positive, negative, and zero synaptic weights using only one control signal. The relationship between synaptic weights and the duration of control signals is also explained in detail. Accordingly, Widrow–Hoff algorithm-based memristive neural network (MNN) circuits are proposed to solve the recognition of three types of character pictures. The functionality of the proposed configurations is verified using SPICE simulation.

Keywords: artificial neural network (ANN); character picture recognition; memristor; memristive neural network (MNN); synaptic circuit; neural network circuit



Citation: Zhang, X.; Wang, X.; Ge, Z.; Li, Z.; Wu, M.; Borah, S. A Novel Memristive Neural Network Circuit and Its Application in Character Recognition. *Micromachines* **2022**, *13*, 2074. <https://doi.org/10.3390/mi13122074>

Academic Editor: Gunuk Wang

Received: 2 November 2022

Accepted: 19 November 2022

Published: 25 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Currently, researchers are giving considerable attention to the implementation of neural networks on hardware platforms to enhance data processing efficiency. The memristive neural network (MNN) circuit is a hardware system that can incorporate memory and computation. It is suitable for high-speed parallel computation and solving the efficiency issues driven by the bottleneck of Von Neumann. Thus, the MNN circuit is a potential candidate for the realization of ANN [1–9]. The unique nonvolatile attributes of memristors and synapses are quite comparable in terms of memory characteristics [10]. To express the weight of synapses, memristors can be directly utilized, which further recognize the application of memristors in neural networks.

The essential link in MNN circuit design is the design of the synaptic circuit [11–17]. Four memristor-based synaptic bridge circuits, which can realize positive, zero, and negative weights is reported in [18–20]. In [20,21], it is suggested to use two cross arrays with the same structure, in which two memristors in the same position act as synaptic circuits. The same input signal is applied to two memristor crossed arrays to obtain the output, and then the difference between the memristors is mapped to positive, zero, or negative weights. Then, the neural network circuit based on the memristors is used to realize character recognition. In [22–24], differential input signals were applied to two rows of a memristor cross array. The sum of the output voltages was expressed as the difference between the resistance values of two memristors, thus obtaining positive, zero, and negative weights. However, when using the array as a synaptic circuit, selecting a certain row or column of the array is necessary, which leads to the inability to realize parallel programming of synaptic circuits in the whole network during operation, which limits the development of accelerated calculation of neural network. In [25], four Metal-Oxide-Semiconductor

(MOS) transistors and a complementary resistance switch were used to form a memory cell. Only positive voltage was used to adjust the resistance, thus simplifying the power supply design and making the control circuit easier to realize. However, this circuit needs two kinds of control signals to adjust the memristance. A 1T2M (one MOS transistor and two memristors) structure memristive synapse circuit was reported in [26]. Specifically, the MOS transistor was used as a switch to determine whether the circuit updated the weight or saved the weight, but the circuit could only realize the positive weight. In [27], a 4T2M (four MOS transistors and two memristors) structure memristive synapse circuit was designed, which required two different control voltages to control the weights in the circuit. In [28], a 4T1M (four MOS transistors, one memristor, and an inverter) structure memristor synapse circuit was designed, which required two kinds of control voltage signals to be applied to the control terminal of the circuit at the same time through an inverter. According to the characteristics of the above circuits, this paper optimizes the above circuit structures and reduces the number of control voltages.

In this paper, a simple memristive synapse circuit is implemented which can adjust the positive, negative, and zero synapse weights through only one control terminal. The memristive neuron circuit is realized by designing a signal summation and activation function circuit. Lastly, through combination with the Widrow–Hoff algorithm, a single-layer neural network circuit is designed to recognize three types of character pictures.

The remainder of this paper is organized as follows: Section 2 presents the design of the memristive synapse circuit along with the description of the circuit weighting operation and weight programming operation; Section 3 presents the design of the neuron circuit, as well as the simulation outcomes ascertaining the function of signal summation and activation; Section 4 shows the character recognition network circuit and confirms the circuit recognition's accuracy using software; lastly, Section 5 summarizes the paper.

2. Design of Memristive Synapse Circuit

2.1. Memristor Model

The HP memristor is a continuous memristor whose resistance changes continuously under the action of applied voltage. It has the advantages of nonvolatility and nanoscale operation, which make it suitable for the design of MNN circuits [3]. The simplified mathematical model of the HP memristor can be expressed as

$$M(q) = R_{\text{ON}} \frac{\mu_v R_{\text{ON}}}{D^2} q(t) + R_{\text{OFF}} \left(1 - \frac{\mu_v R_{\text{ON}}}{D^2} q(t)\right) \quad (1)$$

where $M(q)$ represents the memresistance (Ω), $q(t)$ represents the amount of charge (C) flowing through the memristor, R_{ON} and R_{OFF} represents the resistance of the doped and undoped region, D represents the length (nm) of TiO_2 , and μ represents the average ion mobility (cm^2/Vs) of the material.

2.2. Weighted Operation

Figure 1a presents a new memristive synaptic circuit by applying the HP memristor model. The circuit is composed of four MOS transistors (T_1 – T_4) and two memristors (M_A and M_B) connected in reverse series. Here, T_1 and T_4 are the PMOS transistors, T_2 and T_3 are NMOS transistors, and V_{in} is the input of the synaptic circuit. The control voltage V_G acts on the gates of the four MOS transistors simultaneously to adjust the positive and negative sign of the weight.

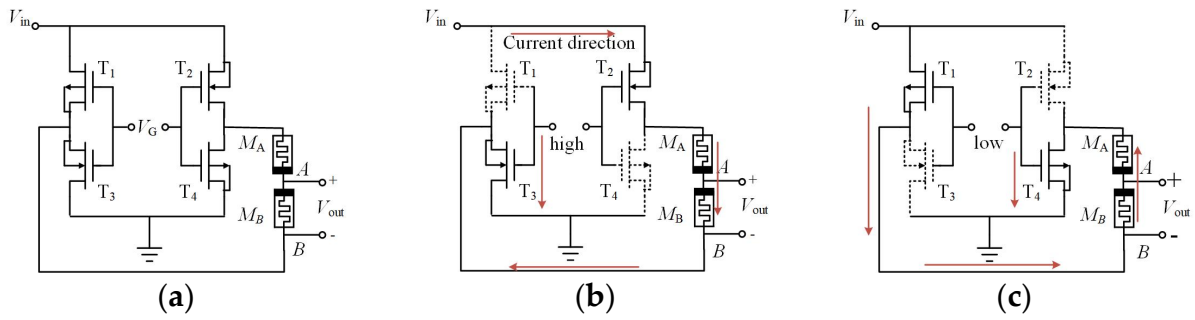


Figure 1. (a) Circuit diagram of the memristive synapse circuit; (b) current path with positive weight; (c) current path with negative weight.

The output voltage, V_{out} is the voltage difference between nodes A and B under the given input, i.e., $V_{out} = V_A - V_B$. Because the two memristors in this circuit are identical and connected in reverse series, the resistance changes of M_A and M_B are always opposite; hence, their sum remains a constant value, i.e., $M_A + M_B = R_{ON} + R_{OFF}$. The sign of the synaptic circuit’s weight can be specified by the control voltage, V_G , as shown in Table 1.

Table 1. Conditions w.r.t the control voltage, V_G .

Conditions	Control Voltage	Transistor State
1	$V_G > 0$	$T_2, T_3 = \text{On}$ $T_1, T_4 = \text{Off}$
2	$V_G < 0$	$T_1, T_4 = \text{On}$ $T_2, T_3 = \text{Off}$

As shown in Table 1 and Figure 1b, *Condition 1* depicts the current path in the circuit. The direction of the current flowing through the memristor M_B is from node A to node B. The output voltage V_{out} is greater than 0 and can be expressed as

$$V_{out} = V_A - V_B = \frac{M_B}{M_A + M_B} V_{in} \tag{2}$$

Likewise, as shown in Table 1 and Figure 1c, *Condition 2* affirms that the current of the memristor M_B flows from node B to node A, and the output voltage V_{out} is negative, as shown below.

$$V_{out} = V_A - V_B = -\frac{M_B}{M_A + M_B} V_{in} \tag{3}$$

Therefore, the following relationship can be observed with respect to the input and output voltages of the synaptic circuit:

$$V_{out} = \omega V_{in} = \frac{V_G}{|V_G|} \cdot \frac{M_B}{M_A + M_B} V_{in} \tag{4}$$

where $\omega = \frac{V_G}{|V_G|} \cdot \frac{M_B}{M_A + M_B}$, for $R_{ON} \ll R_{OFF}$, and the memristances M_A and M_B are all within $[R_{ON}, R_{OFF}]$; thus, when $M_B = R_{ON}$, $\omega = \pm R_{ON} / (R_{OFF} + R_{ON}) \approx 0$, and, when $M_B = R_{OFF}$, $\omega = \pm R_{OFF} / (R_{ON} + R_{OFF}) \approx 1$. It can be concluded that ω can be changed in the range of $[-1, 1]$. Therefore, ω can be used to represent the weight of synaptic circuits and can realize “positive”, “negative”, and “zero” weights.

2.3. Weight Programming Operation

According to the circuit structure and the working mechanism of memristors, the relationship between the weight change of the synaptic circuit and the action time t of the programming voltage can be analyzed. Concretely, because the two memristors are connected in reverse series, when the programming voltage $V_p = +5\text{ V}$ is applied to the input of the synaptic circuit, the changes in $M_A(t)$ and $M_B(t)$ are opposite, resulting in the total memristance $M(t) = M_A(t) + M_B(t)$ in the circuit remaining unchanged. Let $M_A(0)$ and $M_B(0)$ represent the initial values of the two memristors; combined with the memristor model in Equation (1) and the control voltage V_G , the memristances of M_A and M_B in the weight programming stage can be obtained as follows:

$$\begin{cases} M_A(t) = M_A(0) - (R_{\text{OFF}} - R_{\text{ON}}) \times k \times q_1(t) \\ M_B(t) = M_B(0) + (R_{\text{OFF}} - R_{\text{ON}}) \times k \times q_2(t) \end{cases} \quad V_G > 0 \quad (5)$$

$$\begin{cases} M_A(t) = M_A(0) + (R_{\text{OFF}} - R_{\text{ON}}) \times k \times q_1(t) \\ M_B(t) = M_B(0) - (R_{\text{OFF}} - R_{\text{ON}}) \times k \times q_2(t) \end{cases} \quad V_G < 0 \quad (6)$$

where R_{OFF} represents the high-resistance state of the memristor, and R_{ON} represents the low-resistance state. $k = \mu_v \times R_{\text{ON}}/D^2$ is a constant. Since the amount of charge flowing through the two memristors in the series circuit is always the same, i.e., $q_1(t) = q_2(t)$, the total memristance $M(t)$ of the synaptic circuit can be further expressed as

$$M(t) = M_A(t) + M_B(t) = M_A(0) + M_B(0) \quad (7)$$

when $V_G > 0$, according to the current flow direction in Figure 1b and Equations (5) and (7), the corresponding weight change can be obtained as follows:

$$\Delta|\omega(t)| = \frac{\Delta M_B(t)}{M_A(t) + M_B(t)} = -\frac{k \times (R_{\text{OFF}} - R_{\text{ON}}) \times \Delta q(t)}{M_A(0) + M_B(0)} = A \times \Delta t > 0 \quad (8)$$

where $A = k \times (R_{\text{OFF}} - R_{\text{ON}}) \times I / (M_A(0) + M_B(0)) = 30.67$ can be obtained by substitute the parameters into the formula, which is a fixed value. As the resistance of M_B is within the range [100, 16K], the maximum range of ΔM_B is 15.9 k Ω ; thus, $\Delta\omega = 15.9\text{ k}\Omega / 16.1\text{ k}\Omega \approx 0.988$. According to Equation (8), the range of Δt can be obtained as [0, 0.032].

Similarly, when $V_G < 0$, the weight change can be obtained as follows:

$$\Delta|\omega(t)| = \frac{\Delta M_B(t)}{M_A(t) + M_B(t)} = -\frac{k \times (R_{\text{OFF}} - R_{\text{ON}}) \times \Delta q(t)}{M_A(0) + M_B(0)} = -A \times \Delta t < 0 \quad (9)$$

According to the above analysis, the weight change of the synaptic circuit at any time in the weight programming stage can be expressed as

$$\Delta|\omega(t)| = \frac{V_G}{|V_G|} \times \frac{\Delta M_B(t)}{M_A(t) + M_B(t)} = \frac{V_G}{|V_G|} \times \frac{k \times (R_{\text{OFF}} - R_{\text{ON}}) \times \Delta q(t)}{M_A(0) + M_B(0)} = A \times \frac{V_G}{|V_G|} \times \Delta t \quad (10)$$

Thus, the weight $\omega(t)$ of the synaptic circuit at any moment can be obtained as follows:

$$\omega(t) = \frac{V_G}{|V_G|} [|\omega(0)| + \Delta|\omega(t)|] = \frac{V_G}{|V_G|} \times |\omega(0)| + A \times \Delta t \quad (11)$$

where $\omega(0) = \frac{V_G}{|V_G|} \times \frac{M_B(0)}{M_A(0) + M_B(0)}$.

Equation (11) presents the linear functional relationship between the synaptic circuit's weight $\omega(t)$ and the action time t of the programming voltage V_p at any moment. According to the positive and negative control voltage V_G , the operation of increasing or decreasing the weight of the synaptic circuit can be realized.

A comparative analysis of the functions of the proposed memristor synaptic circuit with previously reported studies [26–30] is presented in Table 2. The proposed synaptic circuit offers various advantages in terms of the number of control voltages and the weight range, and it provides good linearity in the programming stage. Therefore, the proposed configuration has better operability in the weight programming stage.

Table 2. Comparison of memristive synaptic circuits.

	[26]	[27]	[28]	[29]	[30]	This Work
Input	Voltage	Voltage	Voltage	Current	Voltage	Voltage
Output	Voltage	Current	Current	Voltage	Current	Voltage
Weight scope	+	+, 0, −	+, 0, −	+, 0, −	+	+, 0, −
Weight linearity	Yes	Yes	No	No	No	Yes
Number of memristors	2	2	1	5	1	2
Number of control voltages	1	2	2	2	1	1

3. The Neuron Circuit

3.1. The Neuron Circuit Design

The synaptic and neuron circuits are two basic units in the MNN circuit. An example of the proposed configuration is verified by considering a neuron circuit with two connected synaptic circuits as shown in Figure 2. The left dashed box represents the two memristive synaptic circuits designed in Section 2, and the right dashed box represents the neuron circuit. The potential difference between A_i and B_i ($i = 1, 2$) is obtained using the two subtractors composed of operational amplifiers A1 and A2, and resistors R_{1-8} . Specifically, when R_1-R_8 are equal, $V_{O1} = V_{A1} - V_{B1} = \omega_1 \times V_{in1}$ and $V_{O2} = V_{A2} - V_{B2} = \omega_2 \times V_{in2}$. The operational amplifier A3 and resistor R_{9-13} constitute an in-phase addition circuit. The output voltage is $V_O = V_{O1} + V_{O2} = \omega_1 \times V_{in1} + \omega_2 \times V_{in2}$, when $R_{12} = R_9 = R_{10} = R_{11}$ and $R_6 = R_9 // R_{10}$.

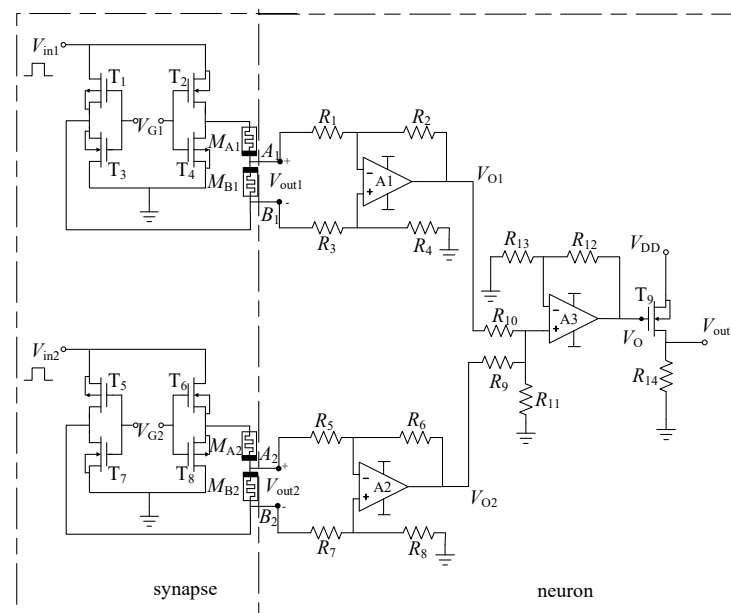


Figure 2. The neuron circuit with two connected synaptic circuits.

Similarly, when a neuron circuit is connected to n synaptic circuits, the relationship between its input and output can be expressed as

$$V_O = \sum_{i=1}^n \frac{V_{Gi}}{|V_{Gi}|} \times \frac{M_{Bi}(t)}{M_{Ai}(t) + M_{Bi}(t)} \times V_{ini} = \sum_{i=1}^n \omega_i(t) \times V_{ini} \tag{12}$$

It can be seen that the neuron circuit realizes the weighted summation of input signals. In Figure 2, the activation function output of the neuron circuit composed of NMOS transistor T_9 and resistor R_{14} is as follows:

$$\text{sign}(V_O) = \begin{cases} 1, & \text{if } V_O \geq 0 \\ 0, & \text{if } V_O < 0 \end{cases} \tag{13}$$

3.2. Simulation Analysis

In the simulation, all the operational amplifiers and MOSFETs are chosen as the universal ones. By considering $R_i = 100 \text{ k}\Omega$ ($i = 1, \dots, 12$), $R_{13} = 50 \text{ k}\Omega$, $R_{14} = 10 \text{ k}\Omega$, $V_{G1} = 5 \text{ V}$, $V_{G2} = -5 \text{ V}$ and the values of HP memristors R_{ON} , R_{OFF} , D , and μ_0 were set to 100Ω , $16 \text{ k}\Omega$, 10 nm , and $10^{-13} \text{ m}^2 \cdot \text{s}^{-1} \cdot \text{V}^{-1}$, respectively. The initial values of M_{A1} and M_{B2} are $16 \text{ k}\Omega$. According to $\omega = \frac{V_G}{|V_G|} \times \frac{M_B}{M_A + M_B}$, it can be known that the initial values of

weights at this time are $\omega_1(0) \approx 0$ and $\omega_2(0) \approx -1$, respectively. Figure 3 shows the simulation results of the LTSpice neuron circuit with two synaptic circuits, where $V_{p1,2}$ are weighted programming voltages applied to the inputs of two synaptic circuits in the programming stage. I_1 and I_2 are the currents of the memristor M_{Bi} ($i = 1, 2$) flowing from the negative electrode to the positive electrode in the two synaptic circuits.

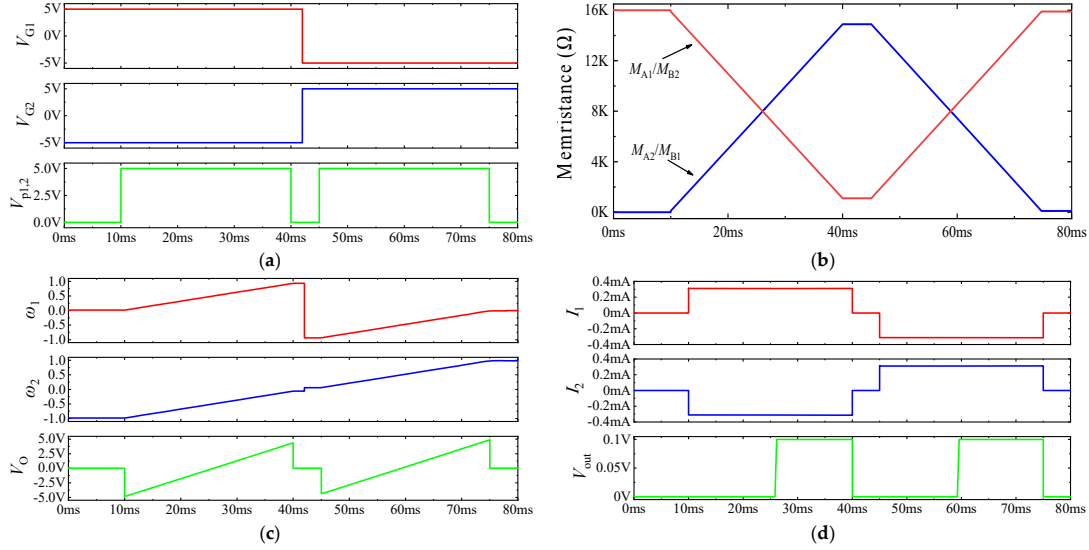


Figure 3. Simulation results of neuron circuit connecting two synaptic circuits: (a) control voltage and weight programming voltage; (b) resistance curve of the memristor in the synaptic circuit; (c) circuit weight and summation voltage V_O ; (d) current flowing through memristor and output voltage V_{out} .

In the time period 0–10 ms, $V_{p1,2} = V_{in1} = V_{in2} = 0$ V, while the memristance and the weights $\omega_1(0) \approx 0$ and $\omega_2(0) \approx -1$ remain unchanged.

The first weight programming stage initiates at 10–40 ms, and the weight programming voltage $V_{p1,2} = 5$ V. According to the initial values of the memristors, since the control voltage V_{G1} of synaptic circuit 1 is kept at 5 V and the initial weight $\omega_1(0) \approx 0$, the initial value of current I_1 can be calculated as 0.31 mA. Likewise, the initial value of current I_2 is -0.31 mA with $V_{G2} = -5$ V, $\omega_2(0) \approx -1$. In this process, according to the initial values of M_{A1} and M_{B2} being 16 kΩ, M_{A2} and M_{B1} are 100 Ω; combined with Equations (5) and (6), the instantaneous resistances of the memristors at each moment can be obtained as follows:

$$\begin{cases} M_{A1}(t) = M_{B2}(t) = 16\text{K} - (16\text{K} - 100) \times 10^5 \times 0.00031(t - 0.01) \\ = 16\text{K} - 4.929 \times 10^5(t - 0.01)(\Omega) \\ M_{B1}(t) = M_{A2}(t) = 100 + (16\text{K} - 100) \times 10^5 \times 0.00031(t - 0.01) \\ = 100 + 4.929 \times 10^5(t - 0.01)(\Omega) \end{cases} \quad (14)$$

At the same time, according to Equations (10) and (11), the weight changes of the two synaptic circuits and the weights at any time can be obtained as follows:

$$\begin{aligned} \Delta|\omega_1(t)| &= \frac{\Delta M_{B1}(t)}{M_{A1}(t) + M_{B1}(t)} = \frac{k \times (R_{OFF} - R_{ON}) \times I_1(t - 0.01)}{M_{A1}(0) + M_{B1}(0)} \\ &= 30.615(t - 0.01) > 0 \\ \Delta|\omega_2(t)| &= \frac{\Delta M_{B2}(t)}{M_{A2}(t) + M_{B2}(t)} = -\frac{k \times (R_{OFF} - R_{ON}) \times I_2(t - 0.01)}{M_{A2}(0) + M_{B2}(0)} \\ &= -30.615(t - 0.01) < 0 \end{aligned} \quad (15)$$

$$\begin{aligned} \omega_1(t) &= \frac{V_{G1}}{|V_{G1}|} [|\omega_1(0)| + \Delta|\omega_1(0.03)|] = \frac{V_{G1}}{|V_{G1}|} [|0| + 30.615(t - 0.01)] = 30.615(t - 0.01) \\ \omega_2(t) &= \frac{V_{G2}}{|V_{G2}|} [|\omega_2(0)| + \Delta|\omega_2(0.03)|] = \frac{V_{G2}}{|V_{G2}|} [|-1| - 30.615(t - 0.01)] = 30.615(t - 0.01) - 1 \end{aligned} \quad (16)$$

Furthermore, the sum of weighted signals V_O and the output voltage V_{out} of the neuron circuit at each time can be derived as follows:

$$V_O = \omega_1 \times V_{in1} + \omega_2 \times V_{in2} = 306.15(t - 0.01) - 5 \quad (17)$$

$$V_{out} = \text{sign}(V_O) = \text{sign}(306.15(t - 0.01) - 5) = \begin{cases} 1, & \text{if } t \geq 26.33 \text{ ms} \\ 0, & \text{if } t < 26.33 \text{ ms} \end{cases} \quad (18)$$

Again, V_{G1} changed from +5 V to -5 V, while V_{G2} changed from -5 V to +5 V in the time period 40–45 ms, and the current path was set in advance for the next weight programming stage. However, since the weight programming voltage $V_{p1,2} = 0$ V at this stage, the resistance of each memristor and the weight ω_i ($i = 1, 2$) in the synaptic circuit remain unchanged, and $V_{out} = 0$ V.

The second weight programming phase starts at 45–75 ms, and the weight programming voltage is $V_{p1,2} = 5$ V. At this time, according to Equation (14), the initial value of the memristor is $M_{B2}(0.045) = M_{A1}$

(0.045) = M_{A1} (0.04) = 1213 Ω , M_{B1} (0.045) = M_{A2} (0.045) = M_{A2} (0.04) = 14,887 Ω . The control voltage V_{C1} of synapse circuit 1 is at a low level, with the initial weight ω_1 (0.045) = -0.918, the control voltage V_{C2} of synapse circuit 2 is at a high level, with the initial weight ω_2 (0.045) = +0.082, and the current I_1 and the current I_2 are -0.31 mA and 0.31 mA, respectively, at 45 ms. The resistances of the memristors are $M_A(t) = M_{B2}(t) = 1213 - 4.929 \times 10^5 (t - 0.045)$ (Ω), $M_{B1}(t) = M_{A2}(t) = 14887 + 4.929 \times 10^5 (t - 0.045)$ (Ω). The weight changes of the two synaptic circuits were $\Delta|\omega_1(t)| = -30.615(t - 0.045)$, $\Delta|\omega_2(t)| = -30.615(t - 0.045)$, $\omega_1(t) = -0.918 + 30.615(t - 0.045)$, $\omega_2(t) = 0.082 + 30.615(t - 0.045)$. The sum of the weighted signals V_O and output voltage V_{out} of each neuron circuit at each time are $V_O = -4.18 + 306.15(t - 0.045)$, $V_{out} = 0.1$ V ($t \geq 58.65$ ms), 0 V ($t < 58.65$ ms).

Converging with Figure 3, the theoretical investigation is completely compatible with the simulation outcomes that establishes the correct actualization and analysis of the proposed neuron circuit. However, in [14–21], the relationship between the acting time of the input signal and the weight change in the synaptic circuit was not given, which is unfavorable for the training and further research on MNN circuits.

4. Circuit Implementation of the Character Recognition Network

On the basis of the above circuits, a neural network circuit based on the memristors was designed to realize the character picture recognition, which can be extended to recognize any group of characters. As shown in Figure 4a, this work utilized three groups of character pictures (z , v , and n) with the resolution of 3×3 as datasets for the ease of simplicity. Each group of pictures includes three standard pictures and 27 noisy pictures. When the circuit training is finished, any z , v , or n character picture inputted into the recognition network circuit in a specific order will be correctly judged by measuring whether the output of the neuron circuit is at a high level.

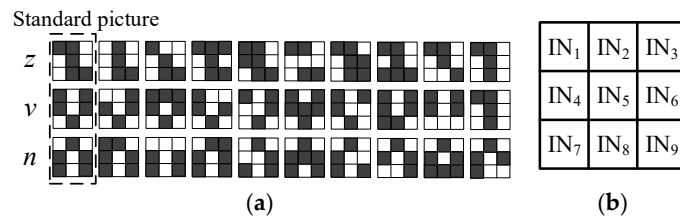


Figure 4. Picture dataset: (a) z , v , and n character pictures; (b) pixel order of pictures.

The three-character recognition network circuit in this paper is composed of three subcircuits with the same structure: Subcircuit z , Subcircuit v , and Subcircuit n . This circuit can also be extended to recognize multiple characters. Different target vectors should be set in the circuit training stage to realize the recognition of each input character. The neuron circuit of Subcircuit z is taken as an example in Figure 5 to exhibit the specific working process of the circuit.

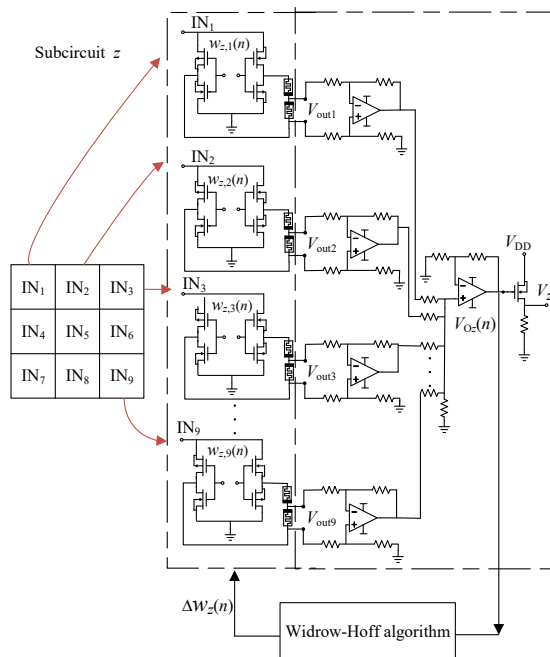


Figure 5. Schematic diagram of neuron circuit of z .

Because the goal of this paper was to identify a character image with a resolution of 3×3 , it was designed to map the pixels in the image to be identified to a one-dimensional vector $In = [IN_1 \ IN_2 \ IN_3 \ IN_4 \ \dots \ IN_9]^T$ according to the sequence shown in Figure 4b and then input it into the nine-input memristive neuron circuit. In this process, logic “1” = 1 V and logic “0” = 0 V were used to represent the black and white in each pixel to realize the picture recognition.

As shown in Figure 5, when certain data in Figure 4a are input into the character recognition network Subcircuit z , the mapped input signal In and the synapse weight ω_z set in the subcircuit are subjected to matrix multiplication operation to obtain the output voltage V_{Oz} in Figure 5, and then the synapse weight is corrected through further training. Lastly, the trained output voltage is sent to the activation function circuit to obtain the output V_z of the subnetwork.

$$V_{Oz}(n) = \omega_z \times In(n) \quad (19)$$

Specifically, the Widrow–Hoff algorithm is used to train the character recognition network circuit. The Widrow–Hoff learning algorithm is an approximate steepest descent method, which uses the mean square error (MSE) as the loss function. Therefore, it is necessary to set the expected outputs of the three subcircuits to t_z , t_v , and t_n , respectively, and then calculate the mean square error with the actual outputs of each subcircuit. Specifically, when the picture z is input to the circuit, $[t_z, t_v, t_n] = [1, 0, 0]$ should be set in the algorithm; that is, the expected output of Subcircuit z is set to 1, and the expected outputs of Subcircuit v and Subcircuit n are both set to 0. When the input pictures are v and n , $[t_z, t_v, t_n] = [0, 1, 0]$ and $[t_z, t_v, t_n] = [0, 0, 1]$ should be selected. Taking Subcircuit z as an example, the error signal in the n iteration is

$$e_z(n) = t_z - V_{Oz}(n) \quad (20)$$

Then, the loss function can be obtained as follows:

$$\theta_z(n) = e_z^2(n) \quad (21)$$

Because neural network learning aims to find a suitable $\omega_z(n)$, the mean square error $\theta_z(n)$ is minimum. Therefore, by using $\theta_z(n)$, the partial derivative of $\omega_z(n)$ is calculated, and, after equating the partial result to zero, the minimum value of $\theta_z(n)$ is obtained. The specific gradient vector equation is as follows:

$$\frac{\partial \theta_z(n)}{\partial \omega_z(n)} = -2e_z(n) \times In^T(n) \quad (22)$$

Finally, the update amount of weight correction is obtained as shown in the following equation:

$$\Delta \omega_z(n) = 2\alpha e_z(n) \times In^T(n) \quad (23)$$

where $\Delta \omega_z(n)$ represents the synaptic circuit weights that need to be updated in the n iteration, and α is the learning rate. The algorithm will be more accurate if the value of α is smaller, but it leads to a slower convergence speed of the algorithm. Therefore, α is selected as 0.1. The calculation process of the Widrow–Hoff algorithm can be realized either using a full circuit [28,31,32] or using a combination of software and hardware [33–35]. In this paper, Matlab software was used to complete the above iterative calculation.

According to the above method, the standard picture of the z character was mapped firstly according to the sequence shown in Figure 4b, and one-dimensional vector $In_z = [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1]^T$ could be obtained. The output terminal V_{Oz} of the neuron circuit was connected to the gate of the NMOS transistor; finally, the output V_z could be obtained through this activation function. Because the expected outputs of the character recognition network circuits Subcircuit z , Subcircuit v , and Subcircuit n were different for different input characters, the corresponding correct subcircuit output voltage was 0.1 V, i.e., logic “1”, and the output voltages of the other two subcircuits were all logic “0” = 0 V, so that it could be correctly judged which of the input characters was “ z ”, “ v ”, and “ n ”.

The process of realizing the Widrow–Hoff algorithm by combining software and hardware was as follows: after the circuit output $V_{Oz}(n)$ was obtained in each iteration, the synaptic weight variation $\Delta \omega_z(n)$ was obtained by external training (Matlab), and the synaptic weight of the circuit was adjusted synchronously. The specific steps were as follows:

Step I: Initialization: Set the initial weights of all synaptic circuits in Subcircuit z to zero, the learning rate $\alpha = 0.1$, the maximum training times $MAX = 50$, $[t_z, t_v, t_n] = [1, 0, 0]$, and the mean square error $\theta_z(n) < 0.005$ as the judgment condition of network training termination.

Step II: The n -th output $V_{Oz}(n)$ of Subcircuit z is taken out and written into Matlab, the error $e_z(n)$ is calculated, and then the mean square error $\theta_z(n)$ is obtained to determine whether to stop the training. Calculate the gradient value of $\theta_z(n)$ and the weight correction $\Delta \omega_z(n)$ corresponding to the input vector.

Step III: Calculate the weight of each synaptic circuit in Subcircuit z , the action time of programming voltage, and combine the positive and negative control voltage to correct the weight of each synaptic circuit in subcircuit z . At this time, the input voltage of each synaptic circuit in each subcircuit is zero, and the control voltage V_G remains unchanged.

In this paper, the character recognition network circuit shown in Figure 5 was simulated using LTSpice software. During each training, by observing whether the output of each subcircuit of the character recognition network was correct, the number of incorrectly recognized pictures of the recognition network circuit was recorded. After the character recognition network circuit was trained, 30 character picture datasets in Figure 4a were inputted into the circuit in a certain order to verify whether all pictures could be correctly recognized. The relationship between the number of incorrectly recognized pictures of three neural network circuits and the training times was obtained as shown in Figure 6. It can be observed that, due to the increase in training time, the number of incorrectly recognized pictures of each subcircuit gradually showed a downward trend.

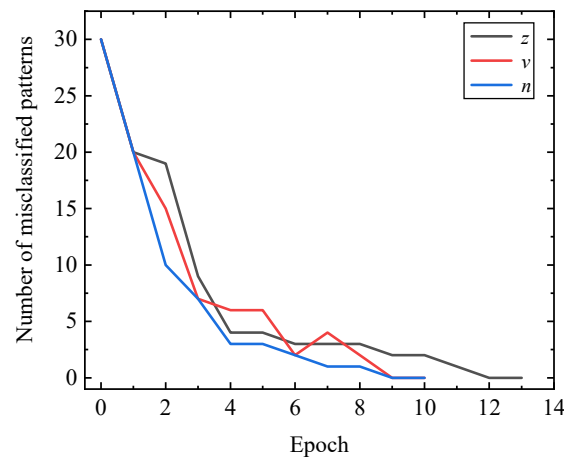


Figure 6. Relationship between the number of misclassifications and training time.

After training the three subcircuits, the accuracy of the identification network circuit was verified. It could be deduced using Equation (10) that the weight change of the synaptic circuit at 0.01 ms was $6.25 \times 10^{-5} \approx 0$, whose effect on the circuit could be ignored. Therefore, it was decided to sequentially input each verification picture at an interval of 0.01 ms. The verification process was divided into four stages, as shown in Table 3.

Table 3. Various stages of verification.

Stages	Time (ms)	Observation
I	0–0.03	Input three standard pictures of <i>z</i> , <i>v</i> , and <i>n</i> into the circuit in turn.
II	0.03–0.05	Continuously input the noisy pictures of each character into the circuit twice in turn
III	0.09–0.12	Continuously input the noisy pictures of each character into the circuit three times in turn
IV	0.18–0.3	Continuously input the noisy pictures of each character into the circuit four times in turn

The simulation results of the circuit verification stages are shown in Figure 7. The green, blue, and red curves represent the output voltages of the subcircuits *z*, *v*, and *n*, respectively. The letters between dotted lines represent the input character pictures at this timepoint. For example, when the picture *z* was input at 0–0.01 ms, the output V_z of the neural network Subcircuit *z* was high (0.1 V). The output of other subcircuits was at a low level (0 V), indicating that the circuit successfully recognized the character picture *z*. As can be seen from Figure 7, when the picture datasets were input into the recognition network circuit in the above order, the circuit could output a corresponding correct waveform. Therefore, the proposed three-character recognition network circuit could correctly recognize all character pictures after training.

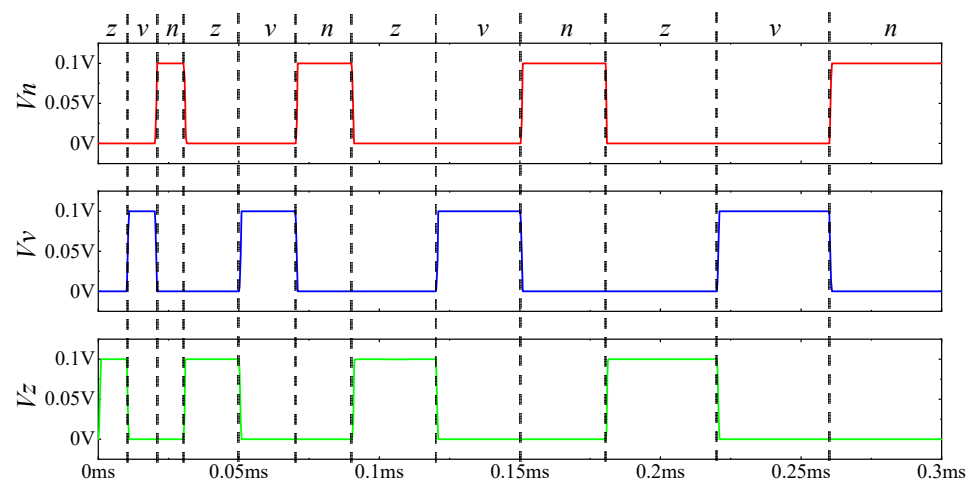


Figure 7. Simulation results of character recognition network circuit and the number of false recognition pictures in the circuit.

5. Conclusions

This paper primarily focused on the application perspective of memristive neural network (MNN) circuits in the direction of character recognition. A new synaptic circuit based on a memristor and CMOS was proposed. On the basis of this synaptic circuit, an MNN circuit based on the Widrow–Hoff algorithm was designed to recognize three kinds of character pictures. The proposed memristive synaptic circuit could only increase or decrease the weight by inputting the digital logic level. Through mathematical derivation, it was observed that the synaptic circuit had good linearity at the weight programming stage. As a function of the structure of the synaptic circuit, positive, zero, and negative weights were realized. Lastly, the proposed character recognition network was simulated on LTSpice, and the accuracy of the circuit was verified. Therefore, the proposed neural network circuit based on memristors is a promising direction for the hardware implementation of ANNs.

Author Contributions: Conceptualization, X.Z., X.W. and Z.G.; methodology, X.Z., X.W. and Z.G.; software, X.Z. and Z.G.; validation, X.Z., X.W., Z.G., M.W. and Z.L.; formal analysis, M.W. and Z.L.; investigation, X.Z., X.W. and Z.G.; resources, X.Z. and Z.G.; data curation, X.Z., X.W. and Z.G.; writing—original draft preparation, X.Z. and Z.G.; writing—review and editing, X.Z., X.W., M.W., Z.L. and S.B.; visualization, S.B.; supervision, X.W.; project administration, X.W.; funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Nature Science Foundation of China (61871429), the Nature Science Foundation of Zhejiang Province (LY18F010012), and the Fundamental Research Funds for the Provincial Universities of Zhejiang (GK219909299001-413).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xu, W.; Wang, J.; Yan, X. Advances in memristor-based neural networks. *Front. Neurosci.* **2021**, *3*, 645995. [[CrossRef](#)]
2. Zhang, X.; Lu, J.; Wang, Z.; Rui, W. Hybrid memristor-CMOS neurons for in-situ learning in fully hardware memristive spiking neural networks. *Sci. Bull.* **2021**, *66*, 1624–1633. [[CrossRef](#)]
3. Strukov, D.; Snider, G.; Stewart, D.; Williams, R. The missing memristor found. *Nature* **2008**, *453*, 80–83. [[CrossRef](#)]
4. Hairong, L.; Chunhua, W.; Cong, X.; Xin, Z.; Herbert, H.C.; Iu, H.H. A Memristive Synapse Control Method to Generate Diversified Multi-Structure Chaotic Attractors. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**. [[CrossRef](#)]
5. Cong, X.; Chunhua, W.; Jinguang, J.; Jingru, S.; Hairong, L. Memristive Circuit Implementation of Context-Dependent Emotional Learning Network and Its Application in Multi-Task. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 3052–3065. [[CrossRef](#)]
6. Leimin, W.; Zhigang, Z.; Ge, M.F. A disturbance rejection framework for finite-time and fixed-time stabilization of delayed memristive neural networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 905–915. [[CrossRef](#)]
7. Leimin, W.; Haibo, H.; Zhigang, Z. Global synchronization of fuzzy memristive neural networks with discrete and distributed delays. *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 2022–2034. [[CrossRef](#)]
8. Junwei, S.; Gaoyang, H.; Zhigang, Z.; Yangfeng, W. Memristor-based neural network circuit of full-function pavlov associative memory with time delay and variable learning rate. *IEEE Trans. Cybern.* **2019**, *50*, 2935–2945. [[CrossRef](#)]
9. Junwei, S.; Juntao, H.; Yanfeng, W.; Peng, L. Memristor-based neural network circuit of emotion congruent memory with mental fatigue and emotion inhibition. *IEEE Trans. Biomed. Circ. Syst.* **2021**, *15*, 606–616. [[CrossRef](#)]
10. Huang, L.; Diao, J.; Nie, H. Memristor based binary convolutional neural network architecture with configurable neurons. *Front. Neurosci.* **2021**, *15*, 639526. [[CrossRef](#)]
11. Yang, X.; Taylor, B.; Wu, A. Research progress on memristor: From synapses to computing systems. *IEEE Trans. Circuits Syst. I-Regul. Pap.* **2022**, *69*, 1845–1857. [[CrossRef](#)]
12. Hong, Q.; Zhao, L.; Wang, X. Novel circuit designs of memristor synapse and neuron. *Neurocomputing* **2019**, *330*, 11–16. [[CrossRef](#)]
13. Ascione, F.; Bianco, N.; Stasio, C.D. Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: A novel approach. *Energy* **2017**, *118*, 999–1017. [[CrossRef](#)]
14. Moussa, H.G.; Hussein, G.A.; Abel-Jabbar, N. Use of model predictive control and artificial neural networks to optimize the ultrasonic release of a model drug from liposomes. *IEEE Trans. Nanobiosci.* **2017**, *16*, 149–156. [[CrossRef](#)] [[PubMed](#)]
15. Del Campo, I.; Echanobe, J.; Bosque, G. Efficient hardware/software Implementation of an adaptive neuro-fuzzy system. *IEEE Trans. Fuzzy Syst.* **2008**, *16*, 761–778. [[CrossRef](#)]
16. Draghici, S. Neural networks in analog hardware—Design and implementation issues. *Int. J. Neural Syst.* **2000**, *10*, 19–42. [[CrossRef](#)]

17. Wei, F.; Chen, G.; Wang, W. Finite-time synchronization of memristor neural networks via interval matrix method. *Neural Netw.* **2020**, *127*, 7–18. [[CrossRef](#)]
18. Adhikari, S.P.; Yang, C.; Kim, H.; Chua, L. Memristor bridge synapse-based neural network and its learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1426–1435. [[CrossRef](#)]
19. Adhikari, S.P.; Kim, H.; Budhathoki, R.K. A circuit-based learning architecture for multilayer neural networks with memristor bridge synapses. *IEEE Trans. Circuits Syst. I-Regul. Pap.* **2014**, *62*, 215–223. [[CrossRef](#)]
20. Prezioso, M.F.; Merrih-Bayat, B.; Hoskins, D. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **2015**, *521*, 61–64. [[CrossRef](#)]
21. Alibart, F.; Zamanidoost, E.; Strukov, D. Pattern classification by memristive crossbar circuits using exsitu and insitu training. *Nat. Commun.* **2013**, *4*, 2072. [[CrossRef](#)]
22. Chabi, D.; Wang, Z.; Bennett, C. Ultrahigh density memristor neural crossbar for on-chip supervised learning. *IEEE Trans. Nanotechnol.* **2015**, *14*, 954–962. [[CrossRef](#)]
23. Hasan, R.; Taha, T.M. Enabling back propagation training of memristor crossbar neuromorphic processors. In Proceedings of the 2014 International Joint Conference on Neural Networks, Beijing, China, 6–11 July 2014; pp. 6–10. [[CrossRef](#)]
24. Hasan, R.T.; Taha, M.; Yakopcic, C. A fast training method for memristor crossbar based multi-layer neural networks. *Analog Integr. Circuits Process.* **2017**, *93*, 443–454. [[CrossRef](#)]
25. Lee, S.; Kim, S.; Cho, K.; Kang, S.M.; Eshraghian, K. Complementary resistive switch-based smart sensor search engine. *IEEE Sens. J.* **2014**, *14*, 1639–1646. [[CrossRef](#)]
26. Yang, J.; Wang, L.; Duan, S. An anti-series memristive synapse circuit design and its application. *Sci. Sin. Inf.* **2016**, *46*, 391–403.
27. Dong, Z.; Lai, C.S.; He, Y. Hybrid dual-complementary metal–oxide–semiconductor/memristor synapse-based neural network with its applications in picture super-resolution. *IET Circuits Devices Syst.* **2019**, *13*, 1241–1248. [[CrossRef](#)]
28. Yang, L.; Zeng, Z.; Shi, X. A memristor-based neural network circuit with synchronous weight adjustment. *Neurocomputing* **2019**, *363*, 114–124. [[CrossRef](#)]
29. Kim, H.; Sah, M.P.; Yang, C. Neural synaptic weighting with a pulse-based memristor circuit. *IEEE Trans. Circuits Syst. I-Regul. Pap.* **2012**, *59*, 148–158. [[CrossRef](#)]
30. Rai, V.K.; Sakthivel, R. Design of artificial neuron network with synapse utilizing hybrid CMOS transistors with memristor for low power applications. *J. Circuits Syst. Comput.* **2020**, *29*, 1–20. [[CrossRef](#)]
31. Soudry, D.; Castro, D.D.; Gal, A.; Kolodny, A.; Kvatinsky, S. Memristor-based multilayer neural networks with online gradient descent training. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2408–2421. [[CrossRef](#)]
32. Ganorkar, S.; Sharma, S.; Jain, A. Soft Computing Algorithms and Implementation on FPGA-A Review. In Proceedings of the International Conference on Smart Data Intelligence, Tiruchirappalli, India, 26 May 2021. [[CrossRef](#)]
33. Ansari, M.; Fayyazi, A.; Kamal, M. OCTAN: An on-chip training algorithm for memristive neuromorphic circuits. *IEEE Trans. Circuits Syst. I-Regul. Pap.* **2019**, *66*, 4687–4698. [[CrossRef](#)]
34. Xu, C.; Wang, C.; Sun, Y. Memristor-based neural network circuit with weighted sum simultaneous perturbation training and its applications. *Neurocomputing* **2021**, *462*, 581–590. [[CrossRef](#)]
35. Bayat, F.M.; Prezioso, M.; Chakrabarti, B. Memristor-based perceptron classifier: Increasing complexity and coping with imperfect hardware. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, Irvine, CA, USA, 13–16 November 2017; pp. 549–554. [[CrossRef](#)]