Research Article

A Novel Chaotic Neural Network Using Memristive Synapse with Applications in Associative Memory

Xiaofang Hu,[1, 2] Shukai Duan,[1] and Lidan Wang[1]

[1] *School of Electronics and Information Engineering, Southwest University, Chongqing 400715, China*
[2] *Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Hong Kong*

Correspondence should be addressed to Shukai Duan, duansk@swu.edu.cn

Chaotic Neural Network, also denoted by the acronym CNN, has rich dynamical behaviors that can be harnessed in promising engineering applications. However, due to its complex synapse learning rules and network structure, it is difficult to update its synaptic weights quickly and implement its large scale physical circuit. This paper addresses an implementation scheme of a novel CNN with memristive neural synapses that may provide a feasible solution for further development of CNN. Memristor, widely known as the fourth fundamental circuit element, was theoretically predicted by Chua in 1971 and has been developed in 2008 by the researchers in Hewlett-Packard Laboratory. Memristor based hybrid nanoscale CMOS technology is expected to revolutionize the digital and neuromorphic computation. The proposed memristive CNN has four significant features: (1) nanoscale memristors can simplify the synaptic circuit greatly and enable the synaptic weights update easily; (2) it can separate stored patterns from superimposed input; (3) it can deal with one-to-many associative memory; (4) it can deal with many-to-many associative memory. Simulation results are provided to illustrate the effectiveness of the proposed scheme.

1. Introduction

Over the last three decades, chaos has been extensively studied by many researchers to understand complex dynamical phenomenon of nonlinear circuits. It has been found that chaotic dynamics may exist in biological neurons in human brain and play important role in associative memory [1, 2]. To mimic biological neuron, a chaotic neuron model has been proposed by Aihara et al. and subsequently many associative chaotic neural network models have been developed by other researchers [3–13].

In the traditional study of associative chaotic neural networks, researchers generally considered that each historic event works on the network with equal degree, which is,

however, hardly the situation in reality. In fact, it has been observed that long-term memory that stores historic events in human brain always decays with time at an exponential rate. Moreover, for traditional chaotic neural network, its structure is very complex, its size is difficult to expand, and its capacity of information processing is limited. Those characteristics pose significant obstacles in CNN's further development. Based on our previous work [10–13], in this paper, we propose a novel associative chaotic neural network with a new exponential decay spatiotemporal effect and a part of spatiotemporal summation whose behavior is more similar to the physiological senses. A memristive neuronal synapse has been designed taking advantage of the variable memristance (resistance of memristor) and memory ability (even if power off) of the memristor [14, 15]. This is expected to simplify the connections between neurons and reduce the complexity of chaotic neural network greatly.

A memristor or memristive device is essentially a basic two-terminal electronic element with nonvolatile continuously variable resistance which can be modulated by the flux or the charge through it. In 1971, Chua theoretically formulated and defined memristor and described that the memristance (resistance of memristor) is defined by the φ-q constitutive relation, that is, $M = d\varphi/dq$ [14]. For about four decades, it did not draw much attention until the recent TiO_2-based nanoscale practical memristor was developed by the HP Labs [16, 17]. Since the startling behavior of metal oxide film as a simple memristor structure was reported, it immediately garnered extensive interests among numerous researchers from academia and industry. Due to the perfect characteristics, it has been proposed in a broad range of applications including resistive random access memory (RRAM) [18, 19], artificial neural networks [20–22], chaotic circuits [23, 24], Boolean logic implementation [25, 26], signal processing, and pattern recognition [27–29]. Here, we show that analog memristive effects can be achieved in a nanoscale TiO_2-based memristor and this type of device exhibits reliable synaptic behavior. It is expected to change the way computer works completely. Moreover, it has potential applications in modern intelligent information processing.

In this paper, we present an implementation scheme for a new chaotic neural network with memristive neural synapse (MCNN). The classical CNN proposed by Aihara et al. is reviewed and the dynamical behaviors for this chaotic neuron are analyzed experimentally in Section 2. Following that the MCNN is presented in Section 3. Firstly, the memristor model is described through mathematical theory, its behaviors are simulated by computer, and its principle characteristics are analyzed in detail. Next, we explain the structure of the MCNN and its dynamical equations also with the simplified memristive synaptic circuit. Associative memory abilities of the MCNN are introduced finally. In order to illustrate the effectiveness of the MCNN, we design a series of computer simulations in Section 4. Reported results indicate that the MCNN can deal with the separation of superimposed patterns, one-to-many and many-to-many association successfully. By integrating the advantages of memristor, chaos, and neural networks, this design will provide theoretical and practical foundations for developing intelligent information processing system.

2. Chaotic Neural Networks with Dynamical Behaviors Analysis

In this section, we briefly review chaotic neural networks. A series of numerical analysis of the firing rate, the bifurcation structure, and the Lyapunov exponent demonstrate the rich dynamical behaviors of the CNN.

2.1. CNN Based on Aihara's Model

The dynamical equation of a chaotic neuron model with continuous functions described by [1],

$$x(t+1) = f\left(A(t) - \alpha \sum_{d=0}^{t} k^d g(x(t-d)) - \theta \right), \tag{2.1}$$

where $x(t+1)$ is the output of the chaotic neuron at the discrete time $t+1$, which is an analogue value between 0 and 1; f is a continuous output function, such as logistic function $f(y) = 1/(1+e^{-y/\varepsilon})$ with the steepness parameter ε; g is a function describing the relationship between the analog output and the refractoriness which can be defined as $g(x) = x$; $A(t)$ is the external stimulation; α, k, and θ are positive refractory scaling parameter, the damping factor, and the threshold, respectively. If defining the internal state $y(t+1)$ as

$$y(t+1) = A(t) - \alpha \sum_{d=0}^{t} k^d g(x(t-d)) - \theta \tag{2.2}$$

we can reduce (2.1) to the following

$$\begin{aligned} y(t+1) &= ky(t) - \alpha g(f(y(t))) + a(t), \\ x(t+1) &= f(y(t+1)), \end{aligned} \tag{2.3}$$

where $a(t)$ is called bifurcation parameter and defined by

$$a(t) = A(t) - kA(t-1) - \theta(1-k). \tag{2.4}$$

The neuron model with chaotic dynamics described above can be generalized as an element of neural network called chaotic neural network (CNN). The dynamics of the ith chaotic neuron with spatiotemporal summation of feedback inputs and externally applied inputs in a CNN composed of M chaotic neurons and N externally applied inputs can be modeled as

$$x_i(t+1) = f(\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)), \tag{2.5}$$

where the external inputs $\xi_i(t+1)$, the feedback inputs $\eta_i(t+1)$, and the refractoriness $\zeta_i(t+1)$ are defined as (2.6)–(2.8), respectively.

$$\xi_i(t+1) = \sum_{j=1}^{N} V_{ij} A_j(t) + k_e \xi_i(t) = \sum_{j=1}^{N} V_{ij} \sum_{d=0}^{t} k_e^d A_j(t-d), \tag{2.6}$$

$$\eta_i(t+1) = \sum_{j=1}^{M} W_{ij} x_j(t) + k_f \eta_i(t) = \sum_{j=1}^{M} W_{ij} \sum_{d=0}^{t} k_f^d x_j(t-d), \tag{2.7}$$

$$\zeta_i(t+1) = -\alpha g\{x_i(t)\} + k_r \zeta_i(t) - \theta_i = -\alpha \sum_{d=0}^{t} k_r^d g\{x_i(t-d)\}, \tag{2.8}$$

where V_{ij} serves as the synaptic weight between the external inputs and the neurons. Similarly, W_{ij} indicates the synaptic weights between two neurons and is trained by Hebbian learning algorithm, such as,

$$W_{ij} = \sum_{k=1}^{M} \left(x_i^k x_j^k \right) \quad \text{or} \quad W_{ij} = \frac{1}{M} \sum_{k=1}^{M} \left(2x_i^k - 1 \right) \left(2x_j^k - 1 \right). \tag{2.9}$$

2.2. Dynamical Behaviors Analysis

The firing rate of a neuron is a fundamental characteristic of the message that conveys to other neurons. It is variable and denotes the intensity of its activation state. As such, it ranges from near zero to some certain maximum depending on its need to convey a particular level of activation. Traditionally, it has been thought that most of the relevant information was contained in the average firing rate of the neuron. The firing rate is usually defined by a temporal average meaning the spikes that occur in a given time window. Division by the length of the time window gives the average firing rate [30]. Here, the average firing rate or excitation number of a single chaotic neuron depicted in (2.3) is defined as

$$\rho = \lim_{n \to +\infty} \frac{1}{n} \sum_{t=0}^{n-1} h(x(t)), \tag{2.10}$$

where h is a transfer function which represents waveform-shaping dynamics of the axon with a strict threshold for propagation of action potentials initiated at the trigger zone and assumed to be $h(x) = 1$ for $x \geq 0.5$ and $h(x) = 0$ for $x < 0.5$. By adjusting the bifurcation parameter $a(t)$ from 0 to 1, when the other parameters are set as $k = 0.7$, $\alpha = 1.0$, $\varepsilon = 0.02$, $y(0) = 0.5$, and the average firing rate is shown in Figure 1.

It is a characteristic of chaotic systems that initially nearby trajectories separate exponentially with time. The Lyapunov exponent or Lyapunov characteristic exponent of a dynamical system is a quantity that characterizes the rate of separation of infinitesimally close trajectories. The rate of separation can be different for different orientations of initial separation vector. Thus, there is a spectrum of Lyapunov exponents—equal in number to the dimensionality of the phase space. It is common to refer to the largest one as the Maximal Lyapunov exponent (MLE), because it determines a notion of predictability for a dynamical
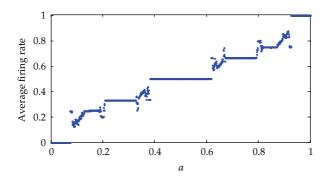
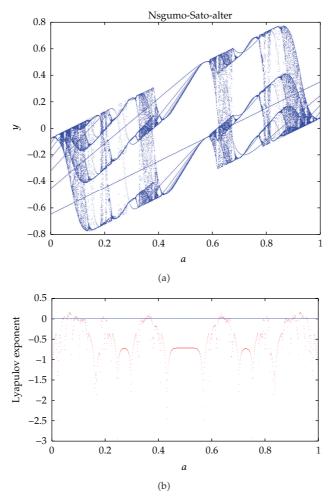**Figure 1:** The average firing rate of the chaotic neuron.

(a)

(b)

Figure 2: Response characteristics diagram of (2.3): (a) the bifurcation structure; (b) the Lyapunov exponent.

system. Specifically, a positive MLE is usually taken as an indication that the system is chaotic [31].

For quantitative analysis of the system (2.3), the Lyapunov exponent λ is defined as follows,

$$\lambda = \lim_{n \to +\infty} \frac{1}{n} \sum_{t=0}^{n-1} \ln \left| \frac{dy(t+1)}{dy(t)} \right|. \tag{2.11}$$

Response characteristics of this system are shown in Figure 2. From Figure 2(a), one can see the bifurcation structure clearly. Figure 2(b) shows the Lyapunov exponent spectrum of this system. The values of the parameters employed are set as the same as those in Figure 1.

3. The Chaotic Neural Network with Memristive Neuronal Synapses

In this section, the memristor is introduced at first to describe its typical characteristics. Then a new chaotic neural network (MCNN) with memristive synapses is presented in detail. Finally, the associative memory abilities of the MCNN model are analyzed.

3.1. Memristor Model Description

All two-terminal nonvolatile memory devices based on resistance switching are memristors, regardless of the device material and physical operating mechanisms [15]. Similar to the resistor (which relates voltage v and current i), capacitor (which relates charge q and voltage v) and inductor (which relates flux φ and current i), the memristor relates the flux φ, and the charge q of a device.

The behaviors of a general memristive device can be described by a pair of coupled equations as following:

$$v = R(w, i)i, \tag{3.1a}$$

$$\dot{w} = f(w, i), \tag{3.1b}$$

where (3.1a) is the i-v equation for the memristive device and w is an (or a group of) internal state variable which is determined by the excitation history of the device. A mathematical model of the typical HP TiO_2-based memristor can be described by [16]

$$v(t) = \left(M_{on} \frac{w(t)}{D} + M_{off} \left(1 - \frac{w(t)}{D} \right) \right) i(t),$$

$$\frac{dw}{dt} = \frac{\mu_v M_{on}}{D} i(t), \tag{3.2}$$

$$w(t) = \frac{\mu_v M_{on}}{D} \int_0^t i(t) dt + w(0) = \frac{\mu_v M_{on}}{D} q(t) + w(0),$$
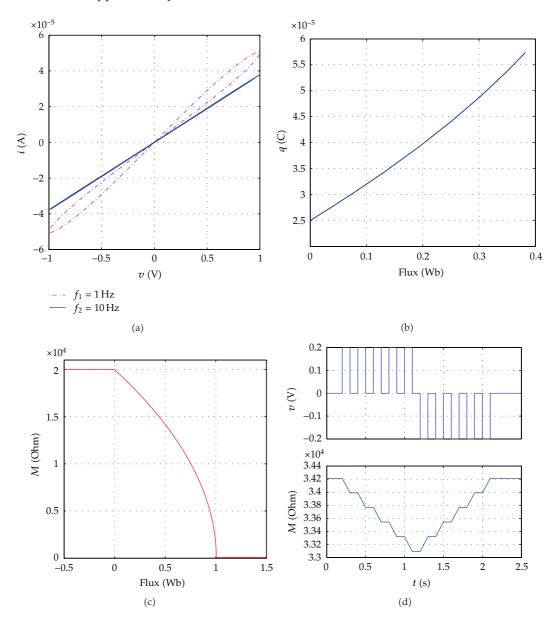
Figure 3: Characteristics of memristor. (a) Typical memristor i-v pinched-hysteresis loop; (b) memristor q-ϕ curve; (c) memristance versus state map; (d) memristance changes with a series of voltages pulses.

where, $M(t) = M_{on}(w(t)/D) + M_{off}(1 - w(t)/D)$ is the total memristance, D and w indicate the thickness of the whole TiO_2 film and the high conductive region, respectively. If $w = 0$ or $w = 1$, memristor reaches the memristance limit value, M_{off} or M_{on}. $w(0)$ denotes the initial value of w. μ_v is average ion mobility parameter in units of $m^2 \, s^{-1} \, V^{-1}$.

Characteristics of the memristor have been analyzed through simulations. By applying a sinusoidal voltage, we can observe the symbolic pinched hysteresis loop (in Figure 3(a)) confined to the first and the third quadrants of the i-v plane. The switching characteristic can be seen from this figure [28], that is, memristor can switch between the high-resistance state

("OFF") and the low-resistance state ("ON"), which has been proposed in binary memory, Boolean calculation, crossbar array, and signal processing abundantly. It should be noted that the pinched hysteresis loop for i-v characteristic implies the device is a memristor, but this curve itself is useless as a model since its contour shape changes with both the amplitude and the frequency of any periodic sine wave like applied voltage or current source and thus cannot be used to predict the current response to arbitrarily applied voltage signals, and vice versa [15]. Moreover, it also can be found that under low-frequency AC excitation, there is an obvious i-v pinched hysteresis loop, but under high-frequency case, the hysteresis loop collapses and memristor degenerates into a fixed resistor.

One of the two ways to predict the response of the device is the q-φ constitutive relation (in Figure 3(b)). As seen in this figure, the memductance $G(\varphi(t_k))$ at the time $t = t_k$ is equal to the slope of the q-φ curve at $\varphi = \varphi(t_k)$, which is called small-signal memductance. Similarly, if the stimulation is a current source, the memristance $M(q(t_k))$ at the time $t = t_k$ is equal to the slope of the φ-q curve at $q = q(t_k)$, and called small-signal memristance. Assuming the constitutive relation curve is continuous at the time $t = t_k$ and the stimulation has sufficiently small amplitude with a fixed frequency, which is called as small-signal condition [15], the small-signal memductance or memristance approximately keeps constant being equal to the curve slope at the time t_k over time and would be indistinguishable from a linear resistance.

Memristance versus state map is an equivalent representation as q-φ curve and also obeys Ohm's Law except that the memristance is not a constant, as illustrated by the example in Figure 3(c). The memristance decreases continuously with the flux increasing, and vice versa. When the flux reaches the limits, minimum or maximum, for a given memristive device, the memristance remain the ultimate value, M_{off} or M_{on}.

Figure 3(d) clearly presents the relationship between the memristance and the applied voltage. By applying a series of small voltage pulses with the same magnitude but opposite polarity, we have observed the stair-like decrement (or increment) of memristance correspondingly. If the bias is a big stimulation, memristance changes continuously. Thus, one can tune and choose different memristances flexibly according to the requirements in different applications by controlling the amplitude, polarity, and duration of the stimulation.

3.2. Memristive Chaotic Neural Network

Based on previous works [10–13, 28], we propose a new chaotic neural network with memristive neuronal synapses (MCNN). This MCNN model has four distinguishing features from the traditional CNN: (1) continuously external input replaces initial input, (2) spatiotemporal effect is replaced with a new exponential decreasing effect, and (3) a part of summation replaces the whole spatiotemporal summation. In traditional associative CNN, researchers considered that each historic effect worked on the network with the same equal level, so they usually assumed decay parameters as constant values which are far from the truth in facts. (4) The memristor is exploited for the synaptic weights. Memristors exhibit dynamical resistance modeled by the stimulation history, which is very similar to the biological neuron synapse and has been proposed in artificial neural network (ANN) as synaptic connection. The MCNN with the features above is more suitable to mimic the biological neuron. Moreover, since the memristor is a nanoscale device and it can simplify the synaptic circuit obviously, the MCNN is expected to be realized in physical implementation as a part of a VSLI chip.
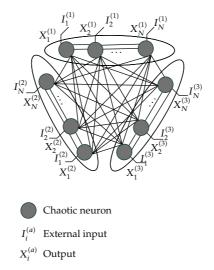
Figure 4: Structure of the memristive chaotic neural networks.

Figure 4 shows the structure of the proposed MCNN model that consists of three layers and all of the neurons are fully connected, which is similar to the Hopfield neural networks.

In the MCNN, the dynamics of the ith chaotic neuron in the αth layer is given as follows:

(1) while $t > t_0$,

$$
x_i{}^\alpha(t+1) = \phi\left(\sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^{L}\sum_{j=1}^{N} W_{ij}^{\alpha\beta}(t) \sum_{d=t-t_0}^{t} k(r1)h_j\left(x_j(t-d)\right) + \sum_{j=1}^{M} W_{ij}^{\alpha\alpha}(t) \sum_{d=t-t_0}^{t} k(r2)h_j\left(x_j(t-d)\right) \right.
$$
$$
\left. + V_i^\alpha(t) \sum_{d=t-t_0}^{t} k(r3)I_i(t-d) - a \sum_{d=t-t_0}^{t} k(r4)g_i(x_i(t-d)) - \theta_i^{(\alpha)} \right),
$$

(3.3)

(2) while $t \leq t_0$, replace $d = t - t_0$ with $d = 0$ in (3.3).

Here, $x_i{}^\alpha(t+1)$ is the output of the chaotic neuron. L, N, and M are the number of layers, the chaotic neurons in the βth layer and in the αth layer; $I_i(t-d)$ is the input of the ith neuron in the αth layer; $V_i^\alpha(t)$ is the connection weight between the input and the neurons in the αth layer; $W_{ij}^{\alpha\beta}(t)$ indicates the weight between the ith chaotic neuron in the αth layer and the jth neuron in the βth layer. The decay parameter and transfer function are defined as $k(u) = e^{-u}$ and $\phi(y) = 2/(1 + e^{-y/\varepsilon})$, respectively. For simplicity, $g(x) = x$ and $h(x) = x$. Moreover, we utilize Hebbian learning regulation as learning algorithm of the synaptic weights.
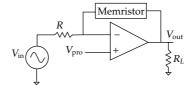
Figure 5: Simplified schematic diagram of the memristive synaptic circuit.

3.3. Memristive Neuronal Synapse in MCNN

We present a simplified memristive synapse circuit using memristors as part of the neuronal synapse taking advantage of the continuously variable memristance and memory ability of memristors. Figure 5 shows the schematic diagram of the memristive synaptic circuit, in which the input voltage source V_{in} is assumed as the output of a neuron, and the output of the synaptic circuit V_{out} is connected to next neuron as the input signal after weighting. The memristor placed in the feedback path can be programmed by the constant programming current using the circuit's voltage to current conversion for a big programming voltage V_{pro}. Assume that the Op-amp is ideal, and then the memristance after programming by a low-frequency voltage V_{pro} for a time width t_p is given by

$$w(t_p) = u_v \frac{M_{on}}{DR} \int_0^{t_p} V_{pro} dt + w(0),$$

$$M(t_p) = M_{off} + (M_{on} - M_{off}) \frac{w(t_p)}{D}. \tag{3.4}$$

The synaptic weight (W_{ij} in (3.3)) is realized by the gain of the amplifier:

$$W_{ij} = \text{Gain} = \frac{v_{out}(t)}{v_{in}(t)} = -\frac{M(t_p)}{R}. \tag{3.5}$$

Different weight values can be obtained easily by changing memristance and the memristance can be set by applying different programming voltages. The synaptic weight keeps constant after the programming operation (the programming voltage cut off) based on the memory ability of memristor. If the synaptic weight should be a positive value, connecting a reversing amplifier to the memristive synaptic circuit may be an effective method.

Next, when the MCNN has been trained by setting its connection weights, it will work normally in some applications such as associative memory. As described in the front, memristors can be considered as a content resistor under a high frequency (or very short operation time) or a small enough external excitation. In this paper, each step of the programs in the simulations uses very short time, about 0.01 s in MATLAB, thus in order to simplify the experiments, we assume that the memristance approximately levels off at the value set in programming stage. At the practical level, for a real memristor, it has some certain threshold voltages presented in literatures [19]. Therefore, in physical implementation of the MCNN, this behavior can be realized by using some voltage-controlling solutions that enable memristance changeable depending on the operation requirement.

3.4. Associative Memory for MCNN

Here, we will primarily analyze the ability of associative memory of the memristive chaotic neural network by undertaking three important applications of the MCNN for examples.

3.4.1. Separation of Superimposed Patterns

Let us assume that a training pattern set $\{X, Y\}$ is memorized in the neural network. When X is given to the network as an externally applied input continuously, the network searches around the input pattern, so pattern X is recalled. When the superimposed pattern $(X + Y)$ is given, the network searches around the X and Y. Due to the property that the chaotic neurons change their states by chaos, the X and the Y can be recalled, respectively. We can also simplify the separation procedure as follows:

$$X + Y \longrightarrow \cdots \longrightarrow X \longrightarrow \cdots \longrightarrow Y \longrightarrow \cdots \longrightarrow X \longrightarrow \cdots . \tag{3.6}$$

3.4.2. One-to-Many Associations

Based on the property of separation of superimposed patterns, assume the following training set (X_1, Y_1, Z_1) has been memorized in a three-layered MCNN, When X_1 is given to the first layer of the network continuously as input, the first layer searches and recalls the X_1 without doubt, the second and the third layer can search around Y_1 and Z_1 by chaotic itinerancy, respectively. So, X_1, Y_1, and Z_1 can be recalled in three layers, respectively, and one-to-many associations is realized.

3.4.3. Many-to-Many Associations

Let us consider the following training set $\{(X_1, Y_1, Z_1), (X_1, Y_2, Z_2), (X_3, Y_3, Z_3)\}$, two patterns of which contain a common term X_1. When X_1 is given to the first layer of the network continuously, the internal state of the chaotic neuron in the second layer is determined by the following formula:

$$\begin{aligned} I^\beta &= X_1 W^{(\alpha\beta)} = X_1 X_1^T Y_1 + X_1 X_1^T Y_2 + X_1 X_3^T Y_3 \\ &= X_1 X_1^T (Y_1 + Y_2) + X_1 X_3^T Y_3, \end{aligned} \tag{3.7}$$

where $W^{(\alpha\beta)}$ is the connection weight matrix between the first layer and the second layer, $X_1 X_3^T Y_3$ is the noise term, and $Y_1 + Y_2$ is the superimposed term resulting from the common term X_1. Similarly, the superimposed term $Z_1 + Z_2$ appears in the third layer. The network searches around Y_1 and Y_2 by chaotic itinerancy in the second layer and searches around Z_1 and Z_2 in the third layer. Then, pattern (X_1, Y_1, Z_1) and (X_1, Y_2, Z_2) can be recalled at different time, realizing the many-to-many associations.

4. Simulation Results

In order to demonstrate the effectiveness and associative memory ability of the proposed MCNN, we perform a series of computer simulations. Since bipolar data (−1 or 1) are used in patterns, the weight values can be calculated by simple derivation based on the Hebbian algorithm formula (the first formula in (2.9)) according to the number of the patterns to be remembered, that is, $W \in \{-N, -N+2, \ldots, 0, \ldots, N-2, N\}$, where W is the weight value and N ($N > 1$) is the number of the patterns. For example, if there are three patterns used in the weight computation at one time, the weight values belong to $\{-3, -1, 0, 1, 3\}$.

In the following experiments, the number of the patterns in the training set used for computing the weights is one, two, and three, respectively. So all of the weight values belong to $\{-3, -2, -1, 0, 1, 2, 3\}$, where, the value "0" means there is no connection between the two neurons, so we do not need to consider it. Moreover, it is known that the memristance must be positive, so the output of the synaptic circuit is negative (seen from (3.5)). Since the weight values X and $-X$ imply the same memristance state but with an opposite weight sign, so we use a reversing amplifier connected to the synaptic circuit to realize the positive weight. Therefore, for a given resistor $R = 10\,k\Omega$, the memristor should be programmed to three states $30\,k\Omega$, $20\,k\Omega$, $10\,k\Omega$, to realize the weights $-3, -2, -1$ (and $3, 2, 1$). Figure 6 shows such three memristance states which we have got and the corresponding voltage pulses (1 V, 0.36 s), (1 V, 1.02 s), (1 V, 1.42 s). Next, by using the synaptic weights obtained from the memristive synaptic circuit, we will study the associative memory ability of the MCNN.

4.1. Separation of Superimposed Patterns of a MCNN

We have designed a MCNN containing 49 neurons to realize the separation of superimposed pattern and the training set is shown in Figure 7. The superimposed pattern $A + B$ is input to the network continuously and the number of output is 20. Network Parameters are set as $a = 9.8$, $kr1 = 0.9$, $kr2 = 0$, $kr3 = 0.1$, $kr4 = 0.9$, $V = 200$, $\theta = 0$, and logistic function with $\varepsilon = 0.02$. This MCNN has a layer and the total valid weight values belong to $\{-3, -1, 1, 3\}$.

Figure 8 shows the output of the MCNN for separation of superimposed pattern, in which pattern A is separated from the superimposed pattern at step 6, 9, 13, and 16, and pattern B is separated at step 5, 8, 11, 14, and 17 successfully. This processing progress takes about 0.01 s in software MATLAB.

4.2. One-to-Many Associations of a MCNN

We have designed a 3-layered MCNN with each layer containing 81 neurons. The parameters are: $a = 10$, $kr1 = 0.95$, $kr2 = 0.1$, $kr3 = 0.1$, $kr4 = 0.95$, $V = 100$, $\theta = 0$, and logistic function with $\varepsilon = 0.015$. The valid weight values in each layer belong to $\{-1, 1\}$ and those between two layers also belong to $\{-1, 1\}$.

Figure 9 shows the training set $(A, a, 1)$. After the whole pattern set is stored in the MCNN by Hebbian learning algorithm, when a pattern A is given to the network as an external input continuously, the whole pattern set can be recalled perfectly (Figure 10) and realize one-to-many associations. This processing progress takes about 0.3 s in software MATLAB.
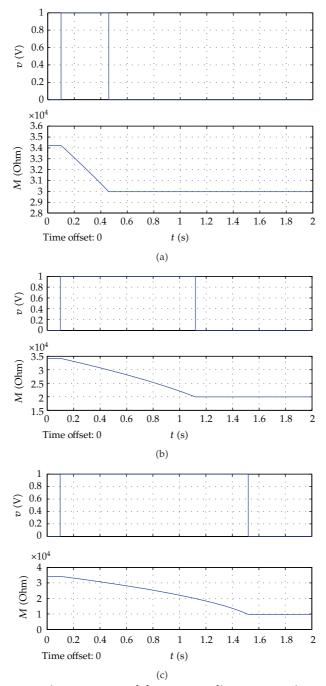
(a)

(b)

(c)

Figure 6: Three memristance states and the corresponding programming voltage pulses.

4.3. Many-to-Many Associations of a MCNN

A 3-layered 81-81-81 MCNN has been designed for many-to-many associations with a part of spatiotemporal summation ($t_0 = 3$ the same as that in the experiments above). The parameters are $a = 10$, $kr1 = 0.95$, $kr2 = 0.1$, $kr3 = 0.1$, $kr4 = 0.95$, $V = 100$, $\theta = 0$,
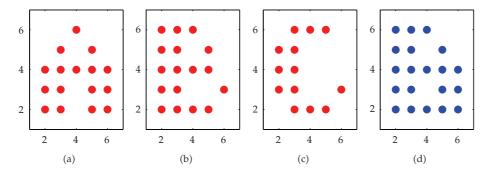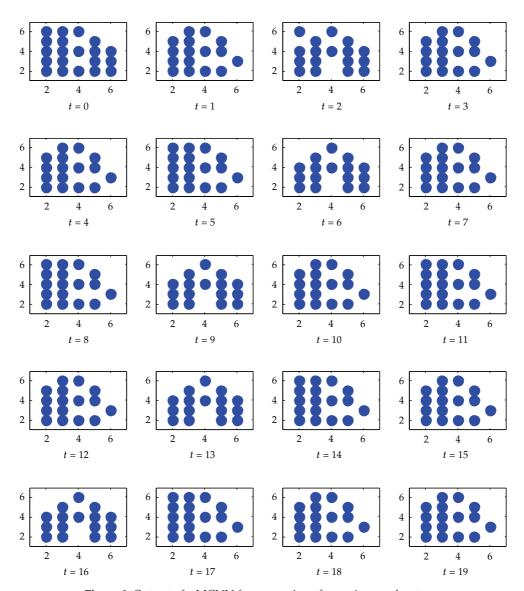
Figure 7: (a)–(c) The training set, (d) the input superimposed pattern.

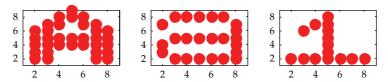**Figure 8:** Output of a MCNN for separation of superimposed pattern.
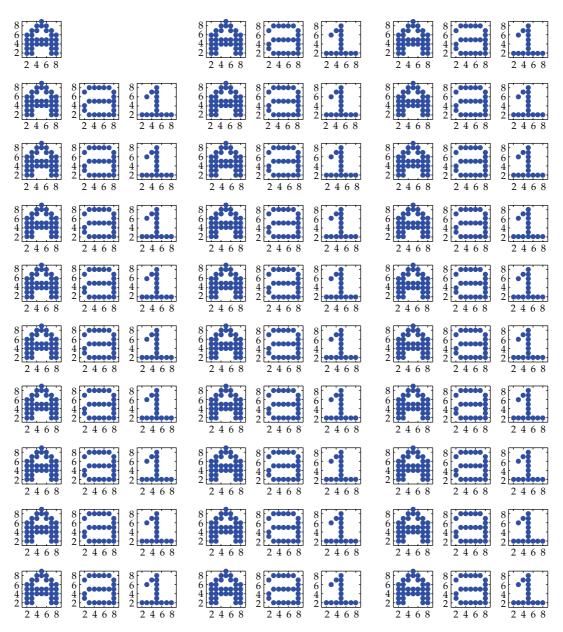
Figure 9: Training set.

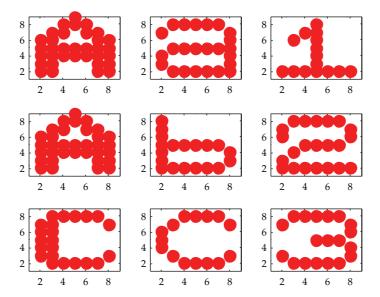**Figure 10:** Output of a MCNN for one-to-many associations.

Figure 11: Training set.

and logistic function with $\varepsilon = 0.015$. The valid weight values in each layer belong to $\{-2, 2\}$ and those between two layers belong to $\{-3, -1, 1, 3\}$.

The training set $\{(A, a, 1), (A, b, 2), (C, c, 3)\}$ shown in Figure 11 is stored in the MCNN, in which the first two patterns include the common term A. The terms A and C are remembered in the first layer, the terms a, b, and c are remembered in the second layer, and 1, 2, and 3 in the third layer. In order to show the outputs clearly, we put the outputs from three layers at the same time together. Figure 12 shows the responses when A is given to the first layer as an external input, in which $(A, a, 1)$ appears 9 times and $(A, b, 2)$ appears 10 times in 30 steps of output. That implies the realization of the many-to-many associative memory of MCNN. This simulation takes about 0.4 s in MATLAB software.

It can be seen from the above experiments that the MCNN can realize associative memory successfully, which indicates it can perform the typical applications as the traditional CNN. Moreover, this MCNN has special characteristics over the traditional ones. First of all, it needs set its connection weights to realize memory ability for a CNN, which is usually called training operation. In order to perform different tasks, these weights should be updated correspondingly. Traditionally, changing these weights is difficult because the synaptic circuit is fixed and complex. In the meantime, this is also a key factor why CNN has not been practically implemented widely. Fortunately, the memristor possesses fast switching (<10 ns), low energy (~1 pJ/operation), high write endurance (10^{10}), multiple-state operation, scalability, synapse-like behaviors, and CMOS compatibility. The size of a memristor is 1/10 of that of a transistor and requires much less energy, so the memristive synaptic circuit would be more simple and smaller than the traditional circuits. What is more important is it can be very easy to update the connection weights just by reprogramming the memristors in the synaptic circuits with proper programming voltages, which saves the big replacement of the synaptic circuit as in traditional CNN and makes the MCNN be expected to be realized in large scale physical circuit.

Figure 12: Output of the proposed MCNN for many-to-many associative memory.

5. Conclusions

An implementation scheme for a new Chaotic Neural Network with memristive neuronal synapses (MCNN) is addressed in this paper. Exponential decay spatiotemporal effect and a part of spatiotemporal summation are considered in the MCNN, which is closer to biological neuron behavior and has higher efficiency in information processing such as associative memory. The characteristics of memristors are analyzed including the *i-v* characteristics,

q-φ constitutive relation, and memristance versus state map with the relationship between the applied voltage and memristance. A memristive synaptic circuit is designed based on a variable gain amplifier. The weight value can be updated easily taking advantage of the variable memristance by programming memristor. When an input signal passes through the memristive synaptic circuit, the weighting operation is realized. Reported simulation results show the effectiveness of the memristive chaotic neural network, that is, it can separate stored patterns from superimposed input pattern and can deal with one-to-many associative memory and many-to-many associative memory perfectly. Furthermore, the memristive synaptic circuit makes the network structure of MCNN more simple and more powerful. At the same time, due to the simple structure (two-terminal), small device size (nano-scale) of the memristor, the proposed MCNN model has a brighter future in physical implementation and holds potential to be implemented in VLSI chips.

Acknowledgments

References

[1] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," *Physics Letters A*, vol. 144, no. 6-7, pp. 333–340, 1990.

[2] Y. Yao and W. J. Freeman, "Model of biological pattern recognition with spatially chaotic dynamics," *Neural Networks*, vol. 3, no. 2, pp. 153–170, 1990.

[3] Y. Osana and M. Hagiwara, "Separation of superimposed pattern and many-to-many associations by chaotic neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 514–519, May 1998.

[4] K. Kaneko, "Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements," *Physica D*, vol. 41, no. 2, pp. 137–172, 1990.

[5] S. Ishii, K. Fukumizu, and S. Watanabe, "A network of chaotic elements for information processing," *Neural Networks*, vol. 9, no. 1, pp. 25–40, 1996.

[6] Y. Osana, M. Hattori, and M. Hagiwara, "Chaotic bidirectional associative memory," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '96)*, pp. 816–821, June 1996.

[7] Y. Osana and M. Hagiwara, "Successive learning in chaotic neural network," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 1510–1515, May 1998.

[8] N. Kawasaki, Y. Osana, and M. Hagiwara, "Chaotic associative memory for successive learning using internal patterns," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 2521–2526, October 2000.

[9] Y. Osana, "Improved chaotic associative memory using distributed patterns for image retrieval," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, pp. 846–851, July 2003.

[10] G. Y. Liu and S. K. Duan, *A Chaotic Neural Network and Its Applications in Separation Superimposed Pattern and Many-to-Many Associative Memory*, vol. 30, Computer Science, Chongqing, China, 2003.

[11] S. Duan, G. Liu, L. Wang, and Y. Qiu, "A novel chaotic neural network for many-to-many associations and successive learning," in *Proceedings of the International Conference on Neural Networks and Signal Processing (ICNNSP '03)*, vol. 1, pp. 135–138, Nanjing, China, December 2003.

[12] S. K. Duan and L. D. Wang, "A novel chaotic neural network for automatic material ratio system," in *Proceedings of the International Symposium on Neural Networks*, Lecture Notes in Computer Science, pp. 813–819, Dalian, China, 2004.

[13] L. Wang, S. Duan, and G. Liu, "Adaptive chaotic controlling method of a chaotic neural network model," in *Proceedings of the 2nd International Symposium on Neural Networks: Advances in Neural Networks (ISNN '05)*, pp. 363–368, June 2005.

[14] L. O. Chua, "Memristor—the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[15] L. Chua, "Resistance switching memories are memristors," *Applied Physics A*, vol. 102, no. 4, pp. 765–783, 2011.

[16] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[17] R. S. Williams, "How we found the missing Memristor," *IEEE Spectrum*, vol. 45, no. 12, pp. 28–35, 2008.

[18] W. Robinett, M. Pickett, J. Borghetti et al., "A memristor-based nonvolatile latch circuit," *Nanotechnology*, vol. 21, no. 23, Article ID 235203, 2010.

[19] P. O. Vontobel, W. Robinett, P. J. Kuekes, D. R. Stewart, J. Straznicky, and R. Stanley Williams, "Writing to and reading from a nano-scale crossbar memory based on memristors," *Nanotechnology*, vol. 20, no. 42, Article ID 425204, 2009.

[20] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[21] T. Masquelier and S. J. Thorpe, "Learning to recognize objects using waves of spikes and spike timing-dependent plasticity," in *Proceedings of the IEEE Word Congress on Computational Intelligence (WCCI '10)*, pp. 2600–2607, Barcelona, Spain, July 2010.

[22] A. Afifi, A. Ayatollahi, F. Raissi, and H. Hajghassem, "Efficient hybrid CMOS-Nano circuit design for spiking neurons and memristive synapses with STDP," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E93-A, no. 9, pp. 1670–1677, 2010.

[23] B. Muthuswamy and P. Kokate, "Memristor-based chaotic circuits," *IETE Technical Review*, vol. 26, no. 6, pp. 417–429, 2009.

[24] W. Sun, C. Li, and J. Yu, "A memristor based chaotic oscillator," in *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS '09)*, pp. 955–957, Milpitas, Calif, USA, July 2009.

[25] S. Shin, K. Kim, and S. M. Kang, "Memristor-based fine resolution programmable resistance and its applications," in *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS '09)*, pp. 948–951, July 2009.

[26] T. Raja and S. Mourad, "Digital logic implementation in memristor-based crossbars," in *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS '09)*, pp. 939–943, July 2009.

[27] "Programmable electronics using memristor crossbars," 2009, http://www.ontheknol.com/front-page-knol/programmable-electronics-using-memristor-crossbars.

[28] X. F. Hu, S. K. Duan, L. D. Wang, and X. F. Liao, "Memristive crossbar array with applications in imageprocessing," *Science China Information Sciences*, vol. 41, no. 4, pp. 500–512, 2011.

[29] T. A. Wey and W. D. Jemison, "Variable gain amplifier circuit using titanium dioxide memristors," *IET Circuits, Devices and Systems*, vol. 5, no. 1, pp. 59–65, 2011.

[30] http://icwww.epfl.ch/~gerstner/SPNM/node6.html.

[31] http://en.wikipedia.org/wiki/Lyapunov_exponent.