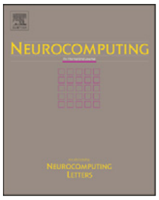
Neurocomputing 74 (2011) 2824–2833

Contents lists available at ScienceDirect

Neurocomputing

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



Design of a chaotic neural network for training and retrieval of grayscale and binary patterns

A. Taherkhani [a],*, S.A. Seyyedsalehi [b], A.H. Jafari [c]

[a] Department of Biomedical Engineering, Amirkabir University of Technology, Tehran 15875-4413, Iran
[b] Department of Biomedical Engineering, Amirkabir University of Technology, Tehran, Iran
[c] Medical Physics & Biomedical Engineering Department, School of Medicine, Tehran University of Medical Sciences, Tehran, Iran

ARTICLE INFO

Article history:
Received 5 January 2010
Received in revised form
13 March 2011
Accepted 20 March 2011
Communicated by R. Kozma
Available online 27 May 2011

Keywords:
Chaotic neural network
Logistic map
Recurrent
Supervisor

ABSTRACT

Experimental and theoretical evidence shows that biological system processing behavior has nonlinear and chaotic properties. The ability of emerging various solutions for a problem and the existence of a supervisor to guide this variety to become close to the goal, are the two main properties of a problem solver. In this paper, a chaotic neural network which uses chaotic nodes with the logistic map as activation functions is designed to make the ability of emerging various solutions and an NDRAM is considered as a supervisor to guide these various solutions. The proposed chaotic neural network has better performance in comparison with Hopfield, NDRAM, and L. Zhao et al. ChNN.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

BRAIN can generate low-amplitude low-frequency fluctuations in electrical and magnetic potential that are detected within the brain and at the surface of the scalp as "brain waves" or the EEG (electroencephalogram). They have the appearance and the statistical properties of random variables in the guise of "colored noise", that is, band-limited white noise; therefore, they are very difficult to read. Recent developments in nonlinear dynamics and chaos theory have proven that EEGs can have deterministic chaos properties [1,2].

Neural networks have chaotic behavior not only on the macroscopic level but also on the microscopic level. Deterministic chaos can naturally generate in a single neuron under some conditions. For example, in the nerve membranes of squid giant axons periodic forcing, such as sinusoidal and periodic pulse stimulation, cause periodic or chaotic responses (depending on parameter values of the frequency and the strength of the force and states of nerve membranes). Chaotic behavior is also reported on the sub-cellular level. Evidences have shown that voltage dependent ion channels in the response to stimuli had random or probably chaotic behavior rather than classical deterministic behavior [3]. These pieces of evidence made the researchers attempt at incorporating the chaos theory into artificial neural networks.

Major contributions to the dynamics of artificial neural networks were made by Cohen and Grossberg [4], Hertz et al. [5], and Hopfield [5,6] in the early 1970s and 1980s, who suggested dynamical behavior in neural networks. A bounded nonlinear dynamical feedback system can become chaotic and, thus, dynamical feedback artificial neural networks can show chaotic behavior [7].

Pinto et al. studied the chaotic behavior of isolated neurons from the stomatogastric ganglion (STG) of the California spiny lobster. They proposed a model that can produce the chaotic behavior of this neuron [8]. Yanga and Yuan considered a parameter in the weight matrix of a simple Hopfield neural network which contained three neurons and determined the variation range of this parameter in which the network had chaotic behavior [9]. In the mentioned models, only the apparent behavior of Chaotic Neural Networks (ChNN) was considered, with little attention to how they can be used for processing tasks.

There are some ChNN with processing ability such as the ChNN proposed by Ishii et al. It is an associative memory system based on parametrically coupled chaotic elements which were used to save and retrieve *N*-dimensional binary vectors [10]. The dynamics of the network of parametrically coupled logistic maps was explored in [11–13]. It was shown that such networks may have enormous memory capacity due to the dense number of different coexisting dynamic attractors. The lattices of chaotic maps were studied by Dmitriev et al. [14], Kaneko and Tsuda [15], and Sinha and Ditto [16]. A chaotic neural network designed

* Corresponding author.
E-mail addresses: A.taherkhani@aut.ac.ir (A. Taherkhani), Ssalehi@aut.ac.ir (S.A. Seyyedsalehi), H_jafari@tums.ac.ir (A.H. Jafari).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.03.037

in [17] to predict time series is capable of oscillating in highly complex aperiodic regime that can also be referred as chaotic regime. Trajectory of the network's state in the state space forms a number of chaotic attractors. The information is then encoded among these attractors. The ability of bifurcating processing units and their networks to rapidly switch between different dynamic modes has been used by Lysetskiy et al. to model new computational properties of neural systems. They have considered a bifurcating neuron based on the control of chaos collapsing to a period-3 orbit in the dynamics of a quadratic logistic map (QLM). Proposed QLM3 neuron is constructed with the third iteration of QLM and uses an external input, which governs its dynamics. The input shifts the neuron's dynamics from chaos to one of the stable fixed points [18]. Zhao et al. have proposed a hybrid model that combines spatial chaotic dynamics and fixed point dynamics for memory storage. Each neuron in the network is a chaotic map (logistic map), whose phase space is divided into two states. One is the periodic dynamic state of period V, representing a V-value retrieved pattern. Another is a chaotic dynamic state corresponding to the searching process [7]. Several researchers such as Kaneko [19,20], Aihara et al. [21], Perrone and Basti [22], Wang [23], Schuster and Stemmler [24], Borisyuk and Borisyuk [25], and finally Minai and Anand [26], have proposed the use of networks of these neurons for pattern classification. In their models, each neuron has chaotic activation function such as Lorenz, Chua, or Rossler [27].

In this paper, a ChNN is designed for training and retrieval of grayscale and binary patterns based on our previous work [28]. The proposed ChNN combines the enormous capability of Nonlinear Dynamic Recurrent Associative Memory (NDRAM) and flexibility of chaotic nodes to achieve a high processing ability. In Section 2, materials and methods involved in the design of the ChNN are introduced. Section 3 contains the simulation results and finally the discussions are put forth in Section 4.

2. Materials and methods

According to [29] a problem solver has two main characteristics:

1. The ability to emerge various approximate solutions for a problem.
2. The existence of a supervisor (or intelligence) to guide this variety to become close to the goal.

In this paper, a ChNN is designed for training and retrieval of grayscale and binary images according to the aforementioned problem solver. A network of chaotic nodes has been considered for input pattern. The chaotic nodes are used to emerge the various possible amounts reconstructing the related pixels in a chaotic manner. A supervisor is designed to conduct the chaotic nodes to approach the goal. In each iteration of recognition procession, the supervisor analyzes the reconstructed pattern which is produced by all chaotic nodes as an input image and uses this global view to conduct chaotic nodes in the next iteration. Thus each chaotic node influences the activity of other nodes via the supervisor and vice versa.

Chaotic nodes network output is used to reconstruct noisy input. $Xn_{[0]}$ is considered as a noisy data produced by adding D noise to a training image Xt as follows:

$$Xn_{[0]} = Xt + D \tag{1}$$

Xt, the original image, can be reconstructed by subtracting D from the noisy image, $Xn_{[0]}$. Chaotic nodes network is used to estimate pixel deviations, D. Chaotic nodes network output, $\Delta X_{[t]}$, is subtracted from the noisy input image in various iterations to reconstruct noisy image as shown in (2):

$$Xn_{[t+1]} = Xn_{[0]} - \Delta X_{[t]} \tag{2}$$

where the components of $\Delta X_{[t]}$ are the chaotic nodes output in the tth iteration, and $Xn_{[t+1]}$ is the reconstructed image in the$(t+1)$th iteration. In each iteration, the supervisor surveys the reconstructed image, $Xn_{[t+1]}$, then it conducts the chaotic nodes to achieve the goal.

The schematic of the proposed ChNN is shown in Fig. 1. The proposed ChNN is composed of two parts: a network of chaotic nodes used for reconstructing the noisy images and a supervisor made of a recurrent neural network. The principle of the chaotic node is described in the next section and the supervisor is introduced in Section 2.2. Section 2.3 describes the training and retrieval stages of the proposed ChNN.

2.1. Principle of the chaotic node

In [7,18,30] a chaotic node was considered for each neuron. Each node activation function is a logistic map. In these models, the output of the ith neuron in the $(t+1)$th iteration was calculated as follows:

$$\Delta x_{i[t+1]} = A_i \Delta x_{i[t]}(1 - \Delta x_{i[t]}) \tag{3}$$

where $\Delta x_{i[t]}$ is the state of ith element in the tth iteration, and A_i is a bifurcation parameter defining the dynamic of ith element. Variation of A_i causes the variation of logistic map dynamic from a fixed point to the periodic behavior or chaotic behavior (Fig. 2). By setting A_i=4, each node has chaotic dynamic. Consequently, each node has the ability for chaotic search in the [0,1] interval.

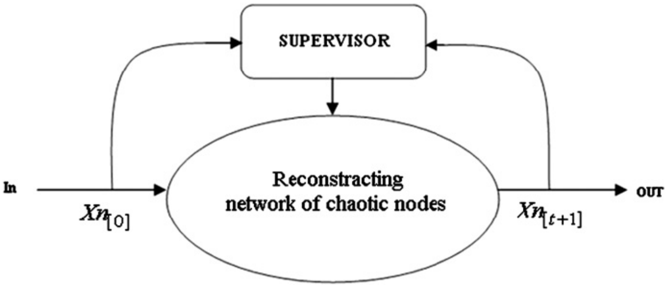


Fig. 1. The proposed ChNN is composed of two parts: a reconstructing network of chaotic nodes emerging various approximate solutions for the problem and a supervisor made of recurrent neural network.

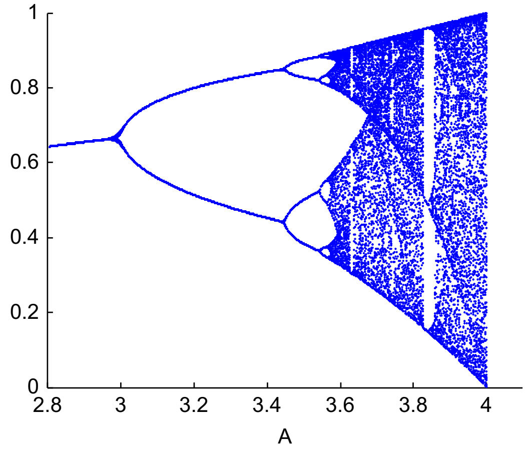


Fig. 2. Bifurcation diagram of the logistic map $\Delta x_{i[t+1]}=A_i\Delta x_{i[t]}(1-\Delta x_{i[t]})$.

Eq. (4) is proposed as a new activation function for each node with the control ability of the searching area:

$$\Delta x_{i[t+1]}=\alpha_i\{A_i[(\Delta x_{i[t]}/\alpha_i+0.5)(0.5-\Delta x_{i[t]}/\alpha_i)]-0.5\} \quad (4)$$

Eq. (4) is a shifted and scale transformed form of 3. The variation region of Δx_i can be controlled by modifying the α_i. α_i is the parameter that can control the searching area of each node in $[-\alpha_i/2,+\alpha_i/2]$ interval. Fig. 3 shows the variation region of Δx_i for two values of α_i. Setting $\alpha_i=1$ in 4 causes the node search in the $[-0.5, +0.5]$ interval, and setting $\alpha_i=0.5$ makes the node explore the $[-0.25, +0.25]$ interval (A_i is considered 4 for each node).

2.2. Supervisor

In the recognition procedure, the noisy input is corrected gradually by a network of chaotic nodes to reconstruct the goal image. The number of chaotic nodes in the network is equal to the number of input pattern pixels. The proposed activation function, (4), is considered for each node. The chaotic nodes evolve chaotically in a proper symmetrical interval around zero. In each iteration, the output of the chaotic nodes network, $\Delta X_{[t]}$, is subtracted from the noisy input, $Xn_{[0]}$, to retrieve the goal image as described by (2). The state of reconstructed pattern, $Xn_{[t+1]}$, is verified by the supervisor in each iteration to estimate its state quality which is called image disagreement. Image disagreement is defined as a criterion to estimate how much a pixel is close to the goal. It is used to guide chaotic nodes to approach the goal. The supervisor has two main duties: 1—for each noisy input, it estimates a proper symmetrical searching interval around zero for each chaotic node, i.e. estimating α_i, 2—in each retrieval iteration, it determines each pixel deviation, $d_{i[t]}$, by estimating image disagreement, $D_{[t]}$, and uses it to conduct related chaotic node.

As shown in Fig. 1, the supervisor made of a recurrent neural network is an essential part of the proposed ChNN. The supervisor produces image disagreement to guide the motion of the chaotic nodes. NDRAM is used to define image disagreement. In this section, NDRAM, chaotic nodes searching area, α, and image disagreement, $D_{[t]}$, which are the supervisor essential elements are introduced.

2.2.1. NDRAM introduction

NDRAM architecture which is proposed by Chartier and Proulx in [31] is shown in Fig. 4. It is an autoassociative and recurrent neural network like the general Hopfield models. The following equation describes its learning rule:

$$W_{[t+1]}=W_{[t]}+\eta(XtXt^T-X_{[t]}X_{[t]}^T) \quad (5)$$

where Xt represents the training input vector, W the weight matrix, $X_{[t]}$ the value of the state vector after iteration t, and η the general learning parameter. If the input vector has N elements, W is an $N\times N$ matrix and NDRAM has N neuron.

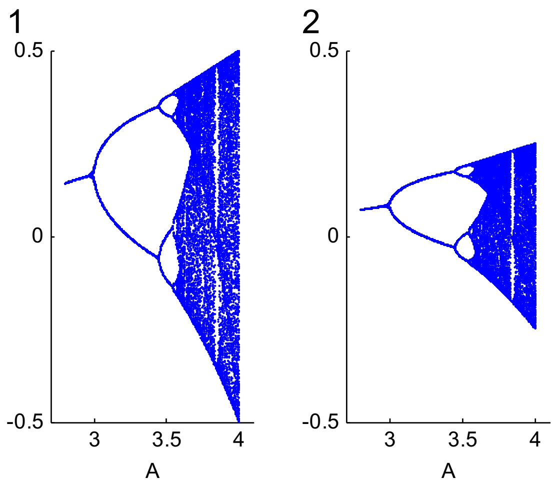


Fig. 3. Variation region of x_i for two different values of α_i in 4: (1) $\alpha_i=1$ cause the neuron explore $[-0.5,+0.5]$, (2) $\alpha_i=0.5$ cause the neuron explore $[-0.25,+0.25]$ ($A_i=4$).

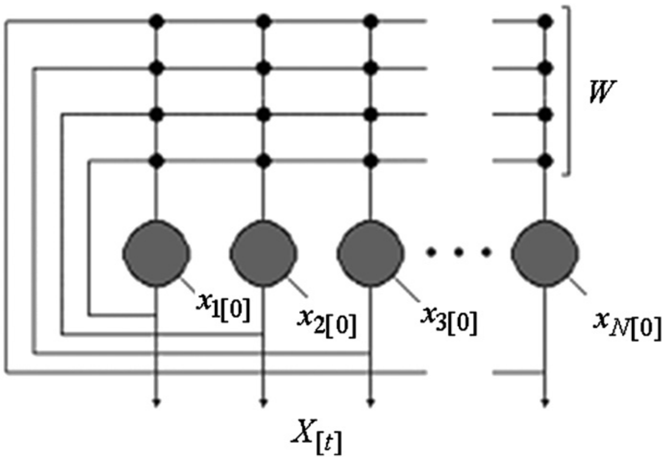


Fig. 4. NDRAM architecture [31].

Eq. (6) is used to calculate $x_{i[t+1]}$, the ith element of $X_{[t+1]}$:

$$\forall i,\ldots,N, x_{i[t+1]}=f(a_i)=\begin{cases}1 & \text{if } a_i>1\\ -1 & \text{if } a_i>1\\ \text{Else} & (\delta+1)a_i-\delta a_i^3\end{cases} \quad (6)$$

where a_i, the ith element of the a, represents the usual activation function ($a=WX_{[t]}$) and δ is the transmission parameter. The transmission parameter (δ) was set to 0.1 and the learning parameter (η) was set to 0.01 according to [31]. To limit the simulation time in 5, the number of transmission iterations before the weight matrix update was set to one ($t=1$). The training and test patterns should be normalized in the interval $(-1, +1)$. The learning was carried out according to the following procedure:

1. Random selection of a training pattern ($X_{[0]}=Xt$).
2. Computing $X_{[t]}$ according to the transmission rule 6.
3. Computing the weight matrix update according to the learning rule 5.
4. Repeating steps 1–3 until the weight matrix converges (around 2000 learning trial).

NDRAM is used to recall the noisy images. Each recall trial is accomplished according to the following procedure:

1. Selecting a training pattern and distorting it by adding a constant distributed noise.
2. Computing $X_{[t+1]}$ according to the transmission rule 6.
3. Repeating step 2 until the state vector stabilizes in an attractor.
4. Repeating steps 1–3 for a different pattern.

2.2.2. Chaotic nodes searching area (α)

In this section, chaotic nodes searching area, α, is described. As shown in 2, a noisy image is reconstructed by subtracting chaotic nodes network output, $\Delta X_{[t]}$, from noisy input image, $Xn_{[0]}$. Thus, chaotic nodes output should be limited in the noise domain. Chaotic nodes searching area, α, which controls the chaotic nodes output according to 4, should be assigned such a value that causes chaotic nodes output fall in the noise domain. A recurrent neural network such as NDRAM can be used to estimate the noise domain for a noisy pattern.

Recurrent neural networks transfer inputs to their corresponding attractors by iteration. The greater difference between an input and its corresponding attractor, the greater distance exists between the input and the output of the recurrent neural network. This property is used to achieve a criterion to estimate how much a pattern closes to its corresponding goal pattern. Thus, it can be used to estimate the amount of the input pattern noise. Mathematical relation can be used to show this property.

It can be shown that the distance between NDRAM noisy input, $X_{[0]}=Xn_{[0]}$, and its first iteration output, $X_{[1]}=f(WXn_{[0]})$, is proportioned to the input pattern noise, D, and it can be used to estimate chaotic nodes searching area, α. Eq. (7) determines the distance between NDRAM noisy input, $X_{[0]}$, and its first iteration output, $X_{[1]}$:

$$X_{[1]}-X_{[0]}=X_{[1]}-Xn_{[0]}=f(WXn_{[0]})-Xn_{[0]} \tag{7}$$

where f is described in (6). The Taylor extension estimation of $f(WXn_{[0]})$ around Xt, training image, in $[-1,1]$ interval is as follows:

$$f(WXn_{[0]})\cong A+B(WXn_{[0]}-Xt) \tag{8}$$

where A is the value of function f in Xt and B is the first derivative of the function f in Xt which are shown in (9) and (10), respectively:

$$A=f(Xt)=\begin{bmatrix} f(xt_1) \\ f(xt_2) \\ \vdots \\ f(xt_N) \end{bmatrix} \tag{9}$$

$$B=f'(Xt)=\begin{bmatrix} f'(xt_1) & 0 & \cdots & 0 \\ 0 & f'(xt_2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & f'(xt_N) \end{bmatrix} \tag{10}$$

where xt_i is the ith component of Xt vector. Eq. (7) is rewrote by using (8) as follows:

$$X_{[1]}-X_{[0]}\cong A+B(WXn_{[0]}-Xt)-Xn_{[0]} \tag{11}$$

Combining (11) and (1) and rearranging the results lead to

$$X_{[1]}-X_{[0]}\cong A+B(WXt-Xt)-Xt+BWD-D \tag{12}$$

Eq. (13) can be obtained from (8) and (12):

$$X_{[1]}-X_{[0]}\cong f(WXt)-Xt+BWD-D \tag{13}$$

Xt is a training data. If NDRAM is well-trained, then $f(WXt)=Xt$ results in

$$X_{[1]}-X_{[0]}\cong BWD-D=(BW-I)D \tag{14}$$

Thus,

$$D\cong (BW-I)^{-1}(X_{[1]}-X_{[0]}) \tag{15}$$

Eq. (15) shows that the difference between input and output of the recurrent neural network, $X_{[1]}-X_{[0]}$, is proportional to the input pattern noise, D. Thus, (16) is supposed to estimate the chaotic node searching area:

$$\alpha=2\left|X_{[1]}-X_{[0]}\right| \tag{16}$$

This leads to each chaotic node with modified logistic map, Eq. (4), explores $[-\alpha_i/2,+\alpha_i/2]$interval chaotically. α_i is ith element of α.

NDRAM surveys all noisy pixels together as an input pattern, $X_{[0]}$, and produces its first iteration output, $X_{[1]}$. This output is used to calculate chaotic nodes searching area, α, by (16). In this way, the activity of each chaotic node, α_i, is influenced by other pixels, which is reconstructed by the activation of other chaotic nodes. Thus, different nodes influence each other via NDRAM activity to produce α (16).

2.2.3. Image disagreement ($D_{[t]}$)

In the proposed ChNN, the supervisor estimates image disagreement and uses it to conduct chaotic nodes in each retrieval iteration. The search for a stored pattern in the Del Moral model is controlled by the level of "disagreement" depending on the network activity and the expected correlations between the nodes that form the connection matrix [30]. In the present study, in each iteration, the deviation of a noisy input image compared to the training image previously stored is estimated by NDRAM. It is used to conduct the chaotic nodes. These deviations are called image disagreement which is proportional to the image noise in each iteration, and referring to (15), it can be calculated by

$$D_{[t]}=\left|X_{[t+1]}-X_{[t]}\right| \tag{17}$$

where $D_{[t]}$ is image disagreement in tth iteration and $X_{[t+1]}$, calculated by (6), is the NDRAM first iteration output in response to $X_{[t]}$.

The supervisor uses image threshold disagreement, ε, to conduct the activation of the chaotic nodes. In each iteration, if $d_{i[t]}$, the ith element of $D_{[t]}$, is less than ε_i, the ith element of ε, this node has reached a proper state and its search will be stopped in the next iteration, otherwise its search will continue. This process continues until the majority or all of the nodes stop their exploration. Disagreement of all the nodes, $D_{[t]}$, is computed in each iteration and if a node is stopped in the previous states, it can continue its search in the next iterations.

ε, image threshold disagreement, is a constant calculated from training image. A set of noisy patterns is constructed by adding noise to the training patterns. Then,ε, image threshold disagreement, is obtained by increasing noise added to the training data until the NDRAM classification performance reaches lower than 100%. The maximum possible noise, for which the NDRAM classification performance is still 100% is used to calculate ε as follows:

$$\varepsilon=2\,\mathrm{mean}\left|Xn_{100\%[0]}-X_{100\%[1]}\right| \tag{18}$$

where $Xn_{100\%[0]}$ is the maximum noisy pattern for which the NDRAM classification performance is 100%, and $X_{100\%[1]}$ is the first iteration of NDRAM output in response to $Xn_{100\%[0]}$which is calculated by (6).

The activation of each chaotic node (α_i,$d_{i[t]}$ and ε_i) is related to the NDRAM output according to (16–18). NDRAM weight matrix interconnects the activation of all nodes reconstructing image pixels. Therefore, different nodes influence other nodes by NDRAM in the supervisor. The cooperation of chaotic nodes as a network and supervisor, which are the essential parts of the proposed ChNN (Fig. 1) for retrieval of noisy patterns, is described in Section 2.3.

2.3. Training and retrieval stage

The Proposed ChNN is used to train and retrieve grayscale and binary patterns. The procedures for training and retrieval grayscale and binary patterns are described separately.

2.3.1. Grayscale image training stage

The network task was to learn 9 correlated grayscale patterns placed on a 32 × 32 grid (Fig. 5). Each pattern is converted into a vector of 1024 components. At training stage, NDRAM is trained by grayscale training images which are noise free as described in Section 2.2.1. The weight matrix is computed and used in the retrieval stage. ε, image threshold disagreement, is calculated by (18).

2.3.2. *Gray scale image retrieval stage*

The proposed ChNN is used for retrieval of Gaussian white noisy grayscale patterns. A Gaussian White noise of mean zero and variance $n/100$ is added to the 32×32 grid training images to produce noisy test data with n percent noise. The training data and its 5% noisy data are shown in Fig. 5.

N (the number of image pixels) chaotic nodes are considered as a network to retrieve the input images. The activation function of each node is given by (4). The α_i(for $i=1,2,\ldots,N$) is ith element of α, chaotic nodes searching area, is calculated by 16. In each iteration, the outputs of chaotic nodes are subtracted from the related pixels to repair noisy input image according to (2). The image disagreement, $D_{[t]}$, is calculated by (17) in each iteration and it is used to conduct chaotic nodes in the next iteration according to (19):

$$\Delta x_{i[t+1]}=\begin{cases}\alpha_i\{4[(\Delta x_{i[t]}/\alpha_i+0.5)(0.5-\Delta x_{i[t]}/\alpha_i)]-0.5\}, & d_{i[t]}>\varepsilon_i\\ \Delta x_{i[t]}, & d_{i[t]}<\varepsilon_i\end{cases}$$
$$\forall i=1,\ldots,N \tag{19}$$

The structure of the proposed ChNN is shown in Fig. 6.

The retrieval stages are as follows:

1. Choosing an image from the test data ($Xn_{[0]}$ which is an $N\times 1$vector).
2. Setting the initial condition of each chaotic node as$\Delta x_{i[0]}=0.001\alpha_i$, in which α_i, ith element of α, is calculated by (16).
3. Reconstructing noisy image by (2).
4. Calculating the image disagreement ($D_{[t]}$) according to (17). For each node, if $d_{i[t]}<\varepsilon_i$ in that $d_{i[t]}$ is the ith component of $D_{[t]}$, then the node search is stopped in the next iteration ($\Delta x_{i[t+1]}=\Delta x_{i[t]}$); otherwise (4) is used to calculate the next iteration for $\Delta x_{i[t+1]}$ (Fig. 6). This process is shown in (19).
5. Repeating stages 3–4 until exploration of the majority or all of the nodes stop.
6. Repeating steps 1–5 for different patterns.

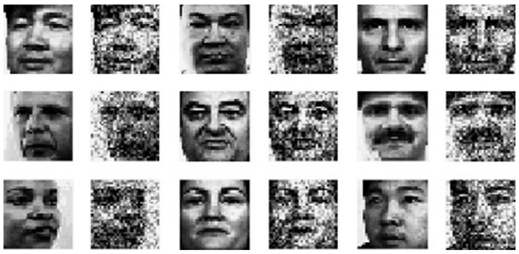


Fig. 5. Nine correlated grayscale training patterns and a corresponding noisy data with 5% Gaussian White noise are shown.

2.3.3. *Binary image training stage*

The proposed ChNN is used to train and to retrieve two binary data sets. The first set is composed of 9 binary patterns, shows in Fig. 7, and the second set contains 26 English alphabets, some of which are shown in Fig. 8. Each pattern contains 32×32 pixels.

The NDRAM is trained by binary images, converted to a range of -1 and $+1$, according to the description in Section 2.2.1. ε, image threshold disagreement, is calculated by (18).

2.3.4. *Binary image retrieval stage*

After the training stage, the proposed ChNN is used for two different recall tasks. The first recall task is to test the network with random pixel flip noise. Various percentages of pixels are randomly selected and then flipped to produce random pixel flip noise. The second recall task is to recall Gaussian White noise pattern. Gaussian White noise of mean zero and variance $n/100$ is added to the training patterns to produce n% noisy images. Fig. 7 shows the first binary training set and its corresponding noisy data, 19.5% of whose pixels are flipped. Fig. 8 shows some data from the second binary training set along with their corresponding noisy patterns; a Gaussian White noise of mean zero and variance $n=0.3$ added to the training patterns.

For binary image retrieval stage Eq. (4) with $\alpha_i=2$ ($i=1,2,\ldots,N$) is considered as the activation function of each node. According to this equation, the node output varies in $(-1,\ 1)$ interval. The binary input patterns, $Xn_{[0]}$, are converted by (20) to produce binary patterns which is coordinated with 2 activation function:

$$xn_{i[0]}=\begin{cases}0.999 & \text{if }\ xn_{i[0]}=1\\ -0.999 & \text{if }\ xn_{i[0]}=-1\end{cases}$$
$$\forall i=1,2,\ldots,N \tag{20}$$

$\Delta X_{[0]}=Xn_{[0]}$ is considered for binary patterns and 4 with $A_i=4$ is used to produce reconstructed pattern. This equation is used to produce various possible goal patterns. Image disagreement, $D_{[t]}$, is calculated in each iteration and is used to conduct the evolution

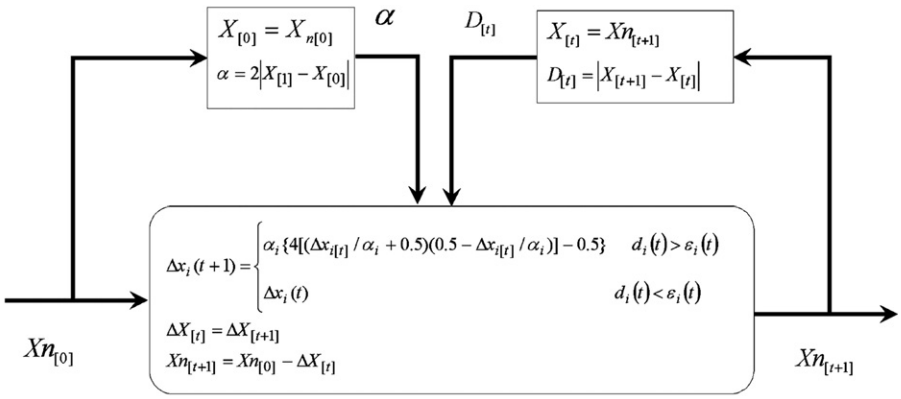

$X_{[0]}=X_{n[0]}$
$\alpha=2|X_{[1]}-X_{[0]}|$
α
$D_{[t]}$
$X_{[t]}=Xn_{[t+1]}$
$D_{[t]}=|X_{[t+1]}-X_{[t]}|$
$\Delta x_i(t+1)=\begin{cases}\alpha_i\{4[(\Delta x_{i[t]}/\alpha_i+0.5)(0.5-\Delta x_{i[t]}/\alpha_i)]-0.5\} & d_i(t)>\varepsilon_i(t)\\ \Delta x_i(t) & d_i(t)<\varepsilon_i(t)\end{cases}$
$\Delta X_{[t]}=\Delta X_{[t+1]}$
$Xn_{[t+1]}=Xn_{[0]}-\Delta X_{[t]}$
$Xn_{[0]}$
$Xn_{[t+1]}$

Fig. 6. The structure of proposed ChNN for grayscale image retrieval stage. α_i(for $i=1,2,\ldots,N$), ith element of α, chaotic nodes searching area, is calculated by 16. Chaotic nodes output are subtracted from the corresponding pixels to repair noisy input image according to 2. The image disagreement, $D_{[t]}$, is calculated by 17 iteratively and is used to conduct chaotic nodes.

of chaotic nodes. The structure of proposed ChNN for the retrieval of binary pattern is shown in Fig. 9.

The retrieval stage of binary image is summarized in the following steps:

1. Choosing a binary pattern from the test data ($Xn_{[0]}$) and converting it by 20. The converted pattern is set as the initial condition for chaotic nodes ($\Delta X_{[0]}=Xn_{[0]}$).
2. Calculating the disagreement of all nodes, $D_{[t]}$, by (17) by considering $\Delta X_{[t+1]}$as reconstructed pattern ($X_{[t]}=\Delta X_{[t+1]}$). For each node if $d_{i[t]} < \varepsilon_i$, its search stops in the next iteration ($\Delta x_{i[t+1]}=\Delta x_{i[t]}$); otherwise 4 is used to calculate the next iteration of $\Delta x_{i[t+1]}$.
3. Repeating step 2 until the majority of nodes stop their search.
4. Repeating steps–3 for different patterns.

3. Results

3.1. Grayscale image

Nine correlated grayscale patterns placed on a 32 × 32 grid are considered (Fig. 5). The proposed ChNN is used to reduce noise in degraded images. Resultant noise-reduced images are applied to a classifier. A two-layer feedforward neural network (FFNN) is used as a classifier.

The classification performance of the proposed ChNN is compared with NDRAM proposed in [31] and FFNN. NDRAM has 1024 fully connected neurons and a 1024 × 1024 weight matrix which is used for training and retrieval of a set of 32 × 32 grayscale patterns. The FFNN which is a two layer neural network has $N=1024$ (number of image pixels) neurons in the first layer and 10 neurons in the output layer of the classifier. This network is trained by noise free grayscale training data which is shown in Fig. 5. The FFNN is created by "newff" MATLAB function. "tansig", Hyperbolic tangent sigmoid transfer function, is used for both input and output layers.

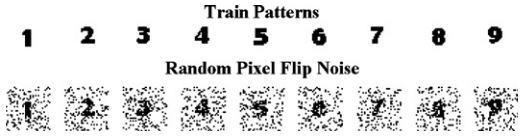


Fig. 7. The first binary training set and its corresponding noisy data, 19.5% of whose pixels are flipped.

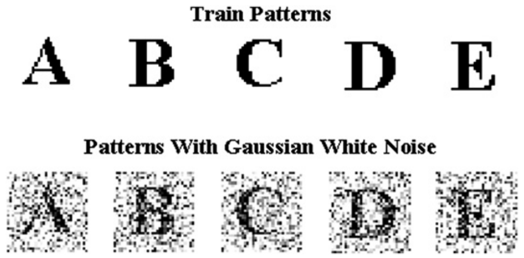


Fig. 8. Some data from the second binary training set with their corresponding noisy patterns; a Gaussian White noise of mean zero and variance $n=0.3$ is added to the training patterns.

Various percentages of Gaussian White noise (described in the previous section) are added to the training data and then three tests are carried out:

1. Reducing the noise by means of the proposed ChNN and classifying the noise-reduced image by the FFNN.
2. Reducing the noise by means of NDRAM and classifying the noise-reduced images by the FFNN.
3. Classifying the noisy images by FFNN.

For each amount of the noise, the performance of different networks is evaluated through thirty trials and, the average performance along with the related standard deviations is shown as the results. The NDRAM classification performance is shown in Fig. 10 for various percentages of Gaussian White noise. A 2% noise is the maximum noise for which the NDRAM classification performance is 100%. Therefore, $Xn_{100\%[0]}$ is a 2% noisy image from which ε can be obtained according to 18. The performance of the proposed ChNN, NDRAM, and FFNN for the retrieval of Gaussian White noisy gray scale images are shown in Fig. 11. The proposed ChNN has very good performance for noises up to 24%. For a 24% noise, the proposed ChNN has 100% classification performance whereas the performance of NDRAM is about 94% (Fig. 11). Two-way ANOVA (analysis of variance) is used to compare the performance of NDRAM and that of the proposed ChNN for the retrieval of Gaussian White noisy gray scale images. The p_value for null hypothesis, H0, that the performance of NDRAM and that

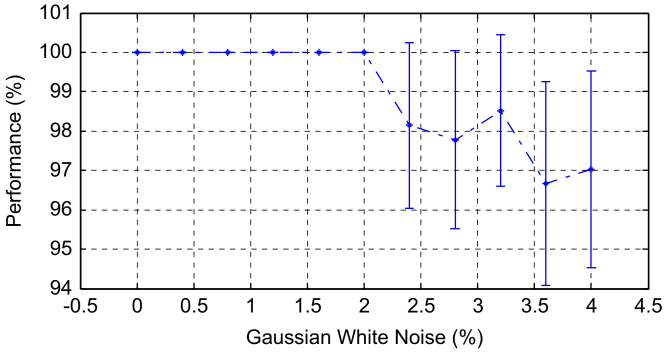


Fig. 10. The NDRAM classification performance for various percentages (%n) of Gaussian White noise. 2% noise is the maximum noise for which the NDRAM classification performance is 100%.

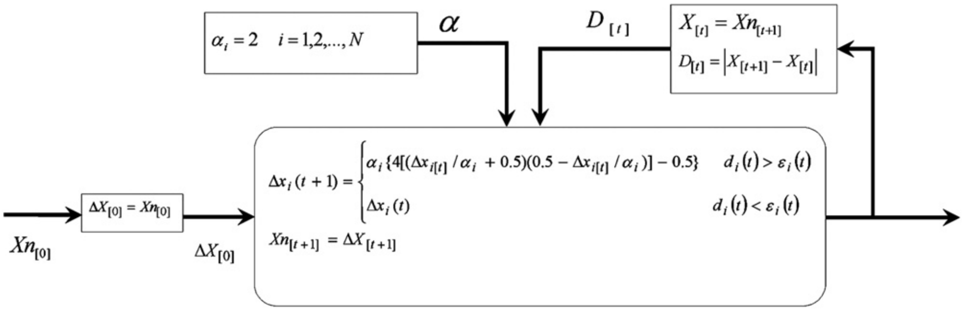

$\alpha_i = 2 \quad i = 1,2,\ldots,N$
α
$D_{[t]}$
$X_{[t]} = Xn_{[t+1]}$
$D_{[t]} = |X_{[t+1]} - X_{[t]}|$
$\Delta x_i(t+1) = \begin{cases} \alpha_i\{4[(\Delta x_{i[t]}/\alpha_i + 0.5)(0.5 - \Delta x_{i[t]}/\alpha_i)] - 0.5\} & d_i(t) > \varepsilon_i(t) \\ \Delta x_i(t) & d_i(t) < \varepsilon_i(t) \end{cases}$
$Xn_{[t+1]} = \Delta X_{[t+1]}$
$\Delta X_{[0]} = Xn_{[0]}$
$Xn_{[0]}$
$\Delta X_{[0]}$

Fig. 9. The structure of proposed ChNN for retrieval of binary patterns. A binary input pattern, $Xn_{[0]}$, converted by 20, is considered as the initial conditions for chaotic nodes, $\Delta X_{[0]}=Xn_{[0]}$. Chaotic nodes produce various possible goal patterns. Image disagreement, $D_{[t]}$, is calculated in each iteration and conducts chaotic nodes evolution.

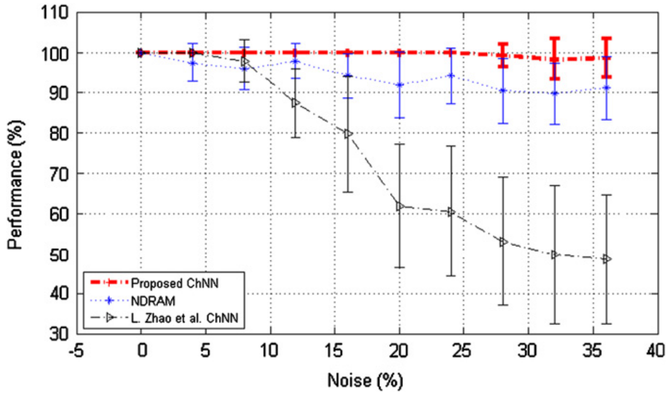


Fig. 11. The performance of the proposed ChNN, NDRAM, and FFNN for the retrieval of Gaussian White noisy gray scale images is shown. The proposed ChNN has very good performance for noises up to 24%. For a 24% noise, the proposed ChNN has 100% classification performance whereas the performance of NDRAM is about 94%.

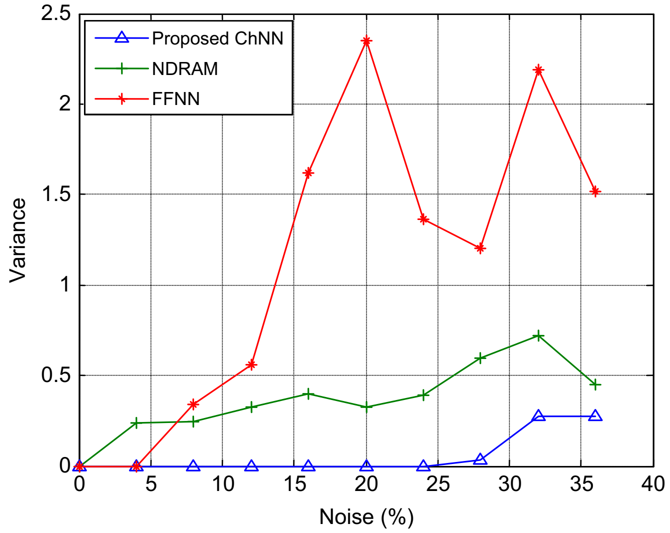


Fig. 12. The standard deviations of the performance of the proposed ChNN, NDRAM and FFNN evaluated through thirty trials is shown for the classification of 32 × 32 grayscale patterns. The standard deviations of the proposed ChNN are less than the others. FFNN has the highest standard deviations.

of the proposed ChNN has same mean, is equal to 0.0000. MATLAB function "anova2" is used to calculate the p_value. The performance of NDRAM and that of the proposed ChNN are significantly different because the p_value is less than 0.01.

The standard deviations of the performance of the proposed ChNN, NDRAM and FFNN evaluated through thirty trials are shown in Fig. 12 for the 32 × 32 grayscale patterns. As shown in this figure, the standard deviations of the proposed ChNN are less than the others. FFNN has the highest standard deviations.

Various reconstructive iterations of five randomly selected pixels ($Xn_{[t+1]}=Xn_{[0]}-\Delta X_{[t]}$) are shown in Fig. 13. As shown in Fig. 13, four corresponding chaotic nodes have become stable and a node has not found its proper state and has continued its chaotic evolution. The recognition procession continues until majority or all of the chaotic nodes become stable. Fig. 14 shows the mean number of chaotic nodes in each iteration for recognition of Gaussian White noisy gray scale images. The number of chaotic nodes reduces in each reconstructive iteration. Fig. 15 shows that the mean value of $D_{[t]}$ is decreased in each iteration and it implies that reconstructed pattern has approached its original pattern.

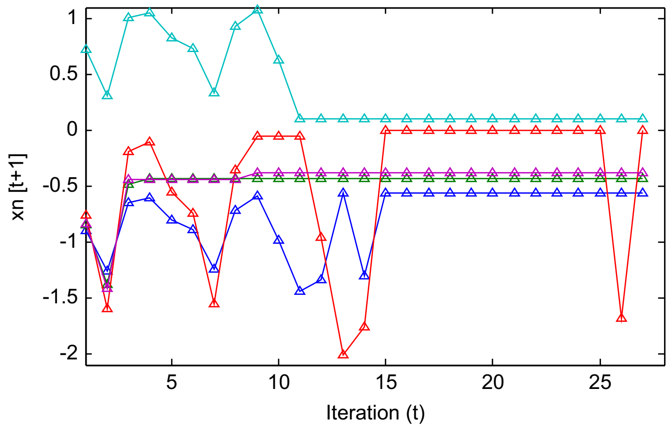


Fig. 13. Various reconstructive iterations of five randomly selected pixels ($Xn_{[t+1]}=Xn_{[0]}-\Delta X_{[t]}$) are shown. Four corresponding chaotic nodes have become stable and a node hasn't found its proper state and has continued its chaotic evolution. The recognition procession continues until majority or all of the chaotic nodes become stable.

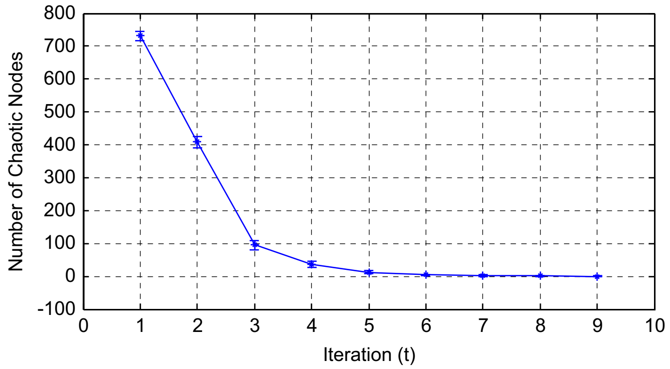


Fig. 14. The mean number of chaotic nodes for recognition of Gaussian White noisy gray scale images reduces in each reconstructive iteration.

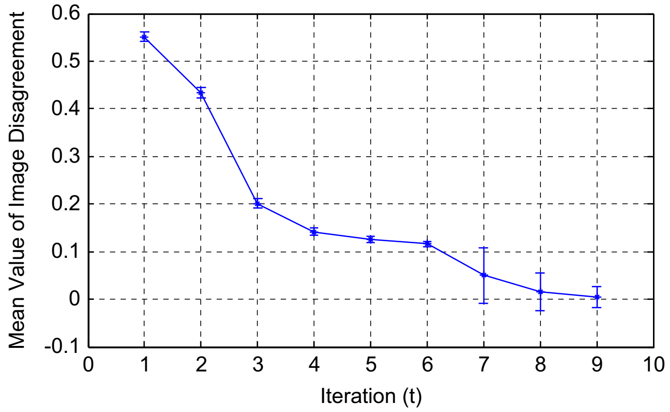


Fig. 15. The mean value of image disagreement, $D_{[t]}$, is decreased in each iteration and it implies that reconstructed pattern has approached its original pattern.

3.2. *Binary image*

Simulation results show that the proposed ChNN has good classification performance on different data sets and on different noises. The performance of the proposed ChNN for the retrieval of binary images is compared with the Hopfield model, NDRAM and L. Zhao et al. ChNN. which is a hybrid model composing the spatial chaotic dynamics for the associative recall to retrieve patterns, similar to Walter Freeman's discovery, and the fixed point dynamics for memory storage, similar to Hopfield and Grossberg's findings [7]. L. Zhao et al. ChNN has an $N \times N$ weight matrix which is calculated by Pseudo-inverse method. N is the number of input image pixels. In this model, chaotic nodes with traditional chaotic equation such as (3) are considered for each pixel. For each node A_i has two amounts. One causes chaotic behavior and the other leads to periodic orbit. The simulation results for two binary training sets described in Section 2.3, and two kinds of noise – Gaussian white noise and random pixels flip noise – are explained below.

Gaussian White binary noisy patterns: A Gaussian White noise of mean zero and variance $n/100$ is added to the binary training images to produce n% Gaussian White noisy data. The resultant noisy images are quantized to make Gaussian White binary noisy patterns. The performance of Hopfield, L. Zhao et al. ChNN, NDRAM, and that of the proposed ChNN for the recognition of these noisy data are compared. Simulations show that the Hopfield model has very poor performance and is not comparable to L. Zhao et al. ChNN, NDRAM, and the proposed ChNN and Figs. 16 and 17 show the performance of L. Zhao et al. ChNN, NDRAM, and that of the proposed ChNN for 0–200 percentage Gaussian White binary noisy patterns of the first data set (1–9 digits) and the second data set (26 English alphabets) separately. Quantization of the analog Gaussian noisy pattern decreases the effect of added noise and leads to high performance. The p_value for null hypothesis, H0, that NDRAM and proposed ChNN has same mean, is less than 0.01. The p_value implies that the performance of the proposed ChNN for the recognition of Gaussian White binary noisy patterns is significantly different from that of NDRAM.

Random pixels flip noisy patterns: Random pixel flip noisy patterns are created by flipping of training pattern pixels. Figs. 18 and 19 show the performance of Hopfield, L. Zhao et al. ChNN, NDRAM, and that of the proposed ChNN for various percentages of pixel flip noisy patterns of the first data set (1–9 digits) and the second data set (26 English alphabets) separately. It is clear that the performance of the proposed ChNN is significantly better than those of the others. The p_value obtained from two-way ANOVA is considerably less than 0.01. It approves that the difference between performances of the proposed ChNN for recognition of various percentages of pixel flip noisy patterns, and that of NDRAM is significant.

Chartier and Proulx in [31] showed that NDRAM has better performance in comparison with those used by Kanter and

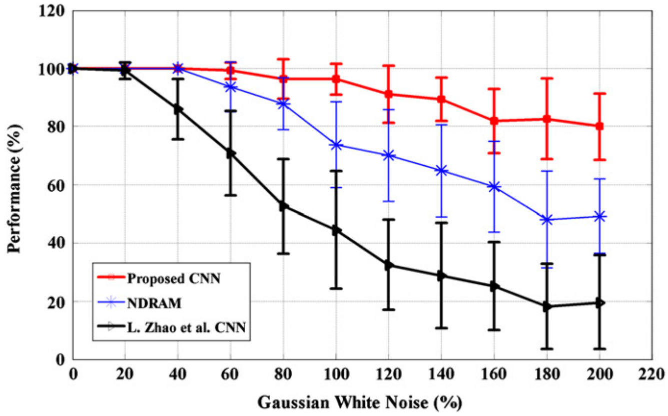


Fig. 16. The performance of L. Zhao et al. ChNN, NDRAM, and that of the proposed ChNN for the recognition of Gaussian White binary noisy patterns of first data set, 1–9 digits, is compared.

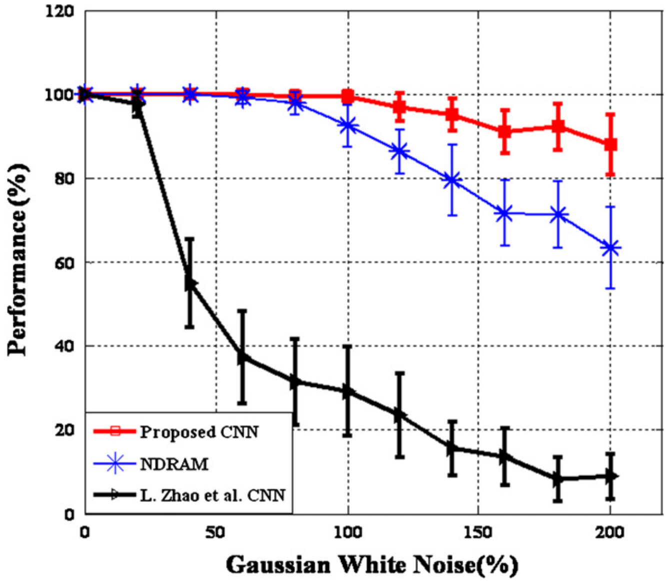


Fig.17. The performance of L. Zhao et al. ChNN, NDRAM, and that of the proposed ChNN for 0–200 percentage Gaussian White binary noisy patterns of the second data set, 26 English alphabets, is shown.

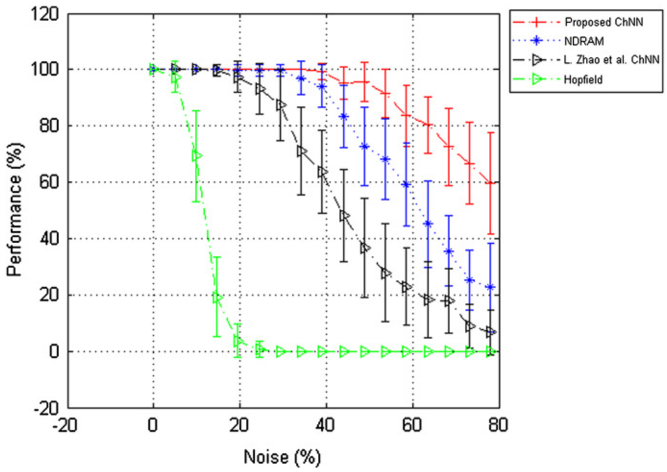


Fig. 18. The performance of Hopfield, L. Zhao et al. ChNN, NDRAM, and that of the proposed ChNN for various percentages of pixels flip noisy pattern of the first data set, 1–9 digits, is shown. It is clear that the performance of the proposed ChNN is significantly better than those of the others.

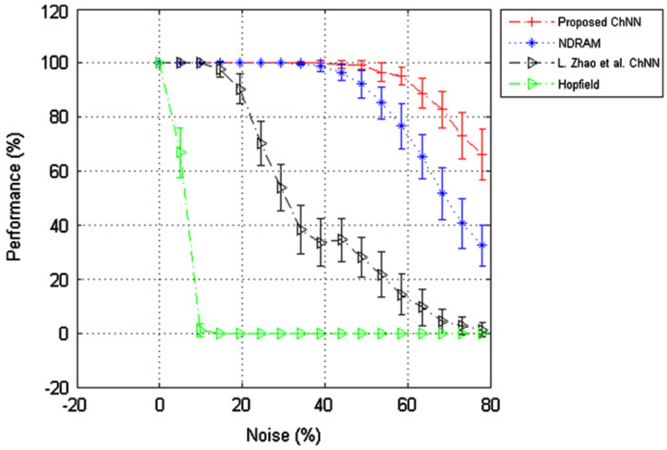


Fig. 19. The performance of Hopfield, L. Zhao et al. ChNN, NDRAM, and the proposed ChNN for various amounts of pixel flip noisy pattern of the second data set, 26 English alphabets, is shown.

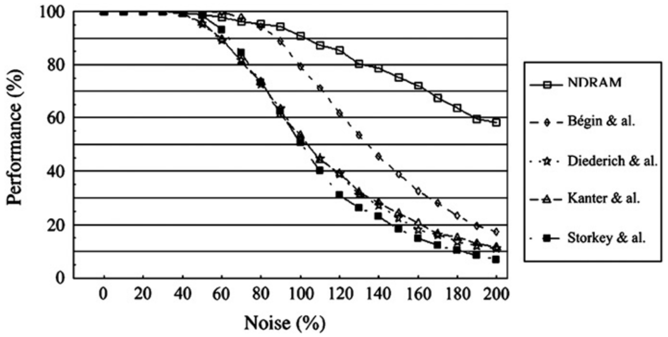


Fig. 20. NDRAM performance percentage in the function of random noise proportion [31] is compared with other neural networks.

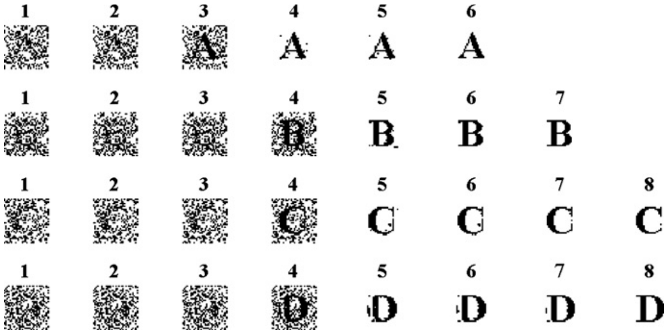


Fig. 21. The retrieval stages of four flipped noisy patterns, which 59% of their pixels are flipped, are shown.

Sompolinsky [32], Storkey and Valabregue [33], Diederich and Opper [34], and Bégin and Proulx [35] (Fig. 20). Therefore, the proposed ChNN has better performance compared with others. Fig. 21 shows the retrieval stages of four flipped pixel noisy patterns, 59% of whose pixels are flipped.

4. Conclusion

Complex systems are composed of a large number of component elements showing interactions between each other. They have a rich dynamics and achieve goals or functions through interactions between elements. Taken separately, elements cannot achieve the same capabilities; the whole can be greater than the sum of the parts [36–38]. Interactions between elements of a complex system trigger the emergence of complex phenomena such as processing intelligence. The proposed ChNN can be considered as a complex system composed of chaotic nodes having a rich dynamics, and they interact between themselves via the supervisor or NDRAM. As a whole, the proposed ChNN, as shown in the previous sections, has a grater processing ability than its elements (chaotic nodes and NDRAM).

Some characteristics that cause the effective performance of a ChNN are as follows:

1. The chaotic property of chaotic nodes causes them not to repetitively search any point in the searching area. The parameters of chaotic nodes can determine the searching area. Proper regulation of the parameters of the nodes makes them search in a specific area properly.
2. Parameter ε, the threshold of image disagreement, gives a proper degree of freedom to the proposed ChNN, which leads to good performance.
3. The searching process continues until almost (not necessarily all) of the nodes stop. In this searching method, the excessive details of the output image are not considered and the searching process stops when a relatively complete image of output appears. In other words, the extra details, which may cause recognition error, are omitted.
4. NDRAM which is used to design the disagreement equation has a good performance in comparison with similar networks (Fig. 20). Therefore, the resultant ChNN has a good processing ability.

Other recurrent neural networks with attractors can be applied to the proposed model and result in an improved neural network. Chaotic nodes and their searching area determined by the parameters of the nodes have an effective impact on the performance of the proposed ChNN. In the present study, a modified logistic map is considered as the activation function of nodes. Other proper chaotic maps can be applied to the model for various problems.

Acknowledgments

The authors appreciate Professor M.R. Hashemi Gholpayeghani and Dr. R. Rahmani for their invaluable comments.

References

[1] W.J. Freeman, Strange attractors that govern mammalian brain dynamics shown by trajectories of electroencephalographic (EEG) potential, IEEE Transactions on Circuits and Systems 35 (7) (1988) 781–783.
[2] G. Lee, N.H. Farhat, The bifurcating neuron network 1, Neural Networks 14 (2001) 115–131.
[3] M.A. Arbib, The Handbook of Brain Theory and Neural Networks, 2nd ed, MIT Press, Cambridge, Massachusetts, London, England, 2003, pp. 205–208.
[4] M.A. Cohen, S. Grossberg, Absolute stability of global pattern formation and parallel memory storage by competitive neural networks, IEEE Transactions on Systems Man Cybernetics 13 (1983) 815–826.
[5] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neurocomputing, Addison-Wesley, Reading, MA, 1991, pp. 1–41.
[6] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences USA 79 (1982) 2554–2558.
[7] L. Zhao, J.C.G. Caceres, A.P.G. Damiance Jr., H. Szu, Chaotic dynamics for multi-value content addressable memory, Neurocomputing 1628–1636 (69, 2006).
[8] R.D. Pinto, P. Varona, A.R. Volkovskii, A. Szűcs, H.D.I. Abarbanel, M.I. Rabinovich, Synchronous behavior of two coupled electronic neurons, Physical Review E (62, 2000) 2644.
[9] Xiao Yanga, Q. Yuan, Chaos and transient chaos in simple Hopfield neural networks, Neurocomputing 69 (2005) 232–241.
[10] S. Ishii, M. Sato, Associative memory based on parametrically coupled chaotic elements, Physica D 121 (3) (1998) 344–366 (23).
[11] N. Farhat, Dynamical networks with bifurcation processing element, in: Proceedings of the NOLTA'97, Honolulu, HI, 1997, pp. 265–268.
[12] N. Farhat, Cortitronics: the way to designing machines with brain-like intelligence, in: Proceedings of the SPIE, vol. 4109, SPIE, Bellingham, Washington, pp. 103–109.
[13] N. Farhat, G.H. Lee, X. Ling, Dynamical networks for ATR, in: Proceedings of the RTO SCI Symposium on Non-Cooperative Air Target Identification Using Radar, Mannheim, Germany, 1998, pp. D1–D4 (published in RTO MP6).
[14] A. Dmitriev, M. Shirokov, S. Starkov, Chaotic synchronization in ensembles of coupled maps, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 44 (10) (1997) 918–926.
[15] K. Kaneko, I. Tsuda, Complex Systems: Chaos and Beyond, Springer, New York, 2000, pp. 57–162.
[16] S. Sinha, W.L. Ditto, Dynamics based computation, Physical Review Letters 81 (10) (1998) 2156–2159.
[17] I. Beliaev, R. Kozma, Time series prediction using chaotic neural networks on the CATS benchmark, Neurocomputing 70 (2007) 2426–2439.
[18] M. Lysetskiy, J.M. Zurada, Bifurcating neuron: computation and learning, Neural Networks (17, 2004) 225–232.
[19] K. Kaneko, Period-doubling of kink–antikink patterns, quasi-periodicity in antiferro-like structures and spatial intermittency in coupled map lattices, Progress in Theoretical Physics (72, 1984) 480–486.
[20] K. Kaneko, Clustering, coding, switching, hierarchical ordering and control in network of chaotic elements, Physica D 41 (1990) 137–142.

[21] K. Aihara, T. Takabe, M. Toyoda, Chaotic neural networks, Physical Letters A 144 (6–7) (1990) 333–340.

[22] A.L. Perrone, G. Basti, Neural images and neural coding, Behavioral and Brain Science 1892 (1995) 368–369.

[23] L.P. Wang, Oscillatory and chaotic dynamics in neural networks under varying operating conditions, IEEE Transactions on Neural Networks 7 (6) (1996) 1382–1388.

[24] H.G. Schuster, & Stemmler, Control of chaos by oscillating feedback, Physical Review E 56 (6) (1997) 6410–6417.

[25] R.M. Borisyuk, G.N. Borisyuk, Information coding on the basis of synchronization of neuronal activity, Biosystems 40 (1–2) (1997) 3–10.

[26] A.A. Minai, T. Anand, Stimulus induced bifurcations in discrete-time neural oscillators, Biological Cybernetics 79 (1) (1998) 87–96.

[27] R. Kozma, W.J. Freeman, Chaotic resonance—methods and applications for robust classification of noisy and variable patterns, International Journal of Bifurcation and Chaos 11 (6) (2001) 1607–1629.

[28] A. Taherkhani, A. Mohamadi, S.A. Seyyedsalehi, H. Davande, Design of chaotic neural network by using chaotic nodes and NDRAM Network, in: Proceedings of the WCCI2008, Hong Kong, 2008, pp. 3500–3504.

[29] F. Heylighen, Self-organization, emergence and the architecture of complexity, in: Proceedings of the AFCET, Paris, 1989, pp. 23–32.

[30] E.D.M. Hernandez, Non-homogenous neural networks with chaotic recursive nodes: connectivity and multi-assemblies structures in recursive processing elements architectures, Neural Networks 18 (2005) 532–540.

[31] S. Chartier, R. Proulx, NDRAM: nonlinear dynamic recurrent associative memory for learning bipolar and nonbipolar correlated patterns, IEEE Transactions on Neural Networks 16 (6) (2005) 1393–1400.

[32] I. Kanter, H. Sompolinsky, Associative recall of memory without errors, Physical Review A vol. 35 (1987) 380–392.

[33] A.J. Storkey, R. Valabregue, The basins of attraction of a new Hopfield learning rule, Neural Networks 12 (1999) 869–876.

[34] S. Diederich, M. Opper, Learning of correlated pattern in spin-glass networks by local learning rules, Physical Review Letters 58 (1987) 949–952.

[35] J. Bégin, R. Proulx, Categorization in unsupervised neural networks: the Eidos model, IEEE Transactions on Neural Networks 7 (1996) 147–154.

[36] R.D. Stacey, D. Griffin, P. Shaw, Complexity and Management, Fad or Radical Challenge to Systems Thinking, 2nd ed, Taylor & Francis e-Library, London, New York, 2001, pp. 17–20.

[37] J. Walleczek, Self-organized Biological Dynamics and Nonlinear Control, first Ed, Cambridge University Press, 2000, pp. 1–12 (Chapter 1).

[38] F. Heylighen, P. Cilliers, C. Gershenson, Complexity and Philosophy, Book Chapter, in: J. Bogg, R. Geyer (Eds.), Complexity, Science and Society, Radcliffe Publishing, Oxford, 2007, pp. 117–134.

Aboozar Taherkhani received the B.Sc. degree in electrical engineering from the Shahid Beheshti University, Tehran, Iran, and M.Sc. degree in Biomedical engineering from the Amirkabir University of Technology, Tehran, Iran, in 2008. His research interests include artificial neural networks, nonlinear dynamical systems, complexity, and nonlinear signal processing.

S.A. Seyyedsalehi received his B.Sc. and M.Sc. degrees both in Electrical Engineering, from Sharif University of Technology, Tehran, Iran in 1982, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 1989 and Ph.D. degree in Biomedical Engineering from Tarbiat Modarres University, Tehran, Iran in 1996. Dr. Seyyedsalehi's research interests are in the areas of speech processing and recognition, biological and artificial neural networks, neural modeling and linear and nonlinear signal processing. He is now an Associate Professor in the Faculty of Biomedical Engineering, Amirkabir University of Technology.

Amir Homayoun Jafari received the B.Sc. degree in Electrical engineering from Sharif University, Tehran, Iran in 1996, the M.Sc. degree in Biomedical engineering from Amirkabir University of technology, Tehran, Iran in 1998 and the Ph.D degree in biomedical engineering from Amirkabir University of technology, Tehran, Iran in 2005. His research interests are modeling, nonlinear control systems, chaotic systems, fuzzy systems and neuromuscular control systems. He is an assistant professor and faculty member of Medical Physics & Biomedical Engineering Department, School of Medicine, Tehran University of Medical Sciences, Tehran, Iran.