Neural Comput & Applic (1999)8:25–32

Neural Computing & Applications

The Object Perceptron Learning Algorithm on Generalised Hopfield Networks for Associative Memory*

J. Ma

Institute of Mathematics, Shantou University, Shantou, Guangdong, P. R. China

We present a study of generalised Hopfield networks for associative memory. By analysing the radius of attraction of a stable state, the Object Perceptron Learning Algorithm (OPLA) and OPLA scheme are proposed to store a set of sample patterns (vectors) in a generalised Hopfield network with their radii of attraction as large as we require. OPLA modifies a set of weights and a threshold in a way similar to the perceptron learning algorithm. The simulation results show that the OPLA scheme is more effective for associative memory than both the sum-of-outer produce scheme with a Hopfield network and the weighted sum-of-outer product scheme with an asymmetric Hopfield network.

Keywords: Associative memory; Hopfield network; Neural network; Perceptron; Radius of attraction; Stable state

1. Introduction

Associative Memories (AMs) have been used for information storage and recall in many applications. Considerable effort has been devoted to the study of AMs. The Hopfield network is an important associative memory model [1], and as proposed in 1982, the sum-of-outer product scheme was applied to store sample patterns. Hopfield demonstrated by computer simulation that a network with n neurons can store about $0.15n$ patterns in the form of its stable states. It is now well known that the memory capacity of a Hopfield network with n neurons is $n/(2\log n)$ patterns, if exact recall is required [2]; and the memory capacity is about $0.15n$ if a little noise is permitted [3].

The Hopfield network is a single layer recurrent network of n bipolar (or binary) neurons uniquely defined by $(\mathbf{W},\theta)$, where $\mathbf{W}$ is a symmetric zero-diagonal real weight matrix, while θ is a real threshold vector. When the weight matrix is changed to an asymmetric and zero-diagonal one, we usually call the network an *asymmetric* Hopfield network. In this paper, we define a generalised Hopfield network as a network with a general (asymmetric or symmetric) zero-diagonal weight matrix. Hereafter, we refer to the Generalised Hopfield Network as GHN.

As a dynamical system, the GHN can also have similar content-addressable memory characteristics to the Hopfield network, especially in randomly asynchronous mode [4]. Therefore, we can apply this kind of neural network to associative memory. Given a sample set $\mathcal{M} = \{X^1,X^2,\ldots,X^m\}$ which consists of m different sample patterns (vectors) in $\{-1,1\}^n$, where

$$X^j = [x_{j,1},x_{j,2},\ldots,x_{j,n}]^{\mathrm{T}} \ (j = 1,2,\ldots,m) \tag{1}$$

The key problem concerning the use of a GHN as an associative memory is how to construct its matrix $\mathbf{W}$ and θ, which enables each of $\{X^1,X^2,\ldots,X^m\}$ to be the stable state of the network with a possibly large basin of attraction. Actually, several learning schemes have already been constructed on GHNs for associative memory [5–9]. Here, we summarise them under two categories.

In one category, the sum-of-outer product scheme is generalised to the weighted one by

$$w_{i,j} = \sum_{k=1}^{m} (1 - \delta_{ij})\lambda_i x_{k,i} x_{k,j} \tag{2}$$

Correspondence and offprint requests to: Jinwen Ma, Institute of Mathematics, Shantou University, Shantou 515063, Guangdong, P. R. China. Email: jwma@mailserv.stu.edu.cu

*This work was supported by China Tianyuan Funds under Project 1937015.

where δ_{ij} is a Kronecker function and λ_i is a weight value to X^i. In 1988, Gardner found a sufficient condition between λ_i and the sample patterns by which the asymmetric Hopfield network (i.e. the GHN) enables each sample pattern to be stable with a nontrivial basin of attraction [5]. She also gave a learning scheme to compute the required **W** and θ from the sample patterns under certain conditions. Then Abbott and Kepler [6] improved the scheme to be more efficient. In 1993 Wang et al [7] proposed another scheme to compute λ_i of Eq. (2) directly using a linear neural network. This kind of scheme has a long learning process. Even if the sample patterns can be stable with a certain basin of attraction in a GHN or linear network using these schemes, their basins of attraction are still as unclear as the desired patterns before the learning process or the resulting ones after the learning process.

In the other category of learning scheme, the spectral (or eigenvector) scheme [8] has been proposed to construct **W** (assuming $\theta = 0$) directly if n is small. The perceptron learning scheme [9] was also proposed to compute **W** and θ directly. These two schemes can make a set of sample patterns stable in a GHN, but they cannot guarantee that each sample pattern has a nontrivial basin of attraction.

In this paper, we combine the ideas of researchers in the two leasing scheme categories, and propose the Object Perceptron Learning Algorithm (OPLA) and OPLA scheme on GHNs for associative memory.

In the following, the absolute radius of attraction is introduced for the stable state of a GHN, and a lower bound of the absolute radius of attraction is obtained in Section 2. Then the OPLA and OPLA scheme are proposed and analysed in Section 3. In Section 4, the simulation experiments are presented to show that the proposed algorithm and scheme are effective for associative memory. A brief conclusion appears in Section 5.

2. The Absolute Radius of Attraction of a Stable State

We first give the mathematical model of a GHN. A GHN with n neurons is uniquely defined by $(\mathbf{W},\theta)$. Here, **W** is an $n \times n$ zero-diagonal real matrix, where element $w_{i,j}$ denotes the weight of the connection from neuron j to neuron i, θ is a vector of dimension n, and where component θ_i denotes the threshold value of neuron i. The state of neuron i at time t is denoted by $x_i(t)$, and it is either '1' or '-1'. The state vector of the GHN at time t is denoted by $X(t) = [x_1(t),x_2(t),\dots,x_n(t)]^T$. When neuron i is now activated, its state at the next time (i.e. $t + 1$) is computed by

$$x_i(t+1) = Sgn(H_i(X(t))) = \begin{cases} 1 & \text{if } H_i(X(t)) \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

where

$$H_i(X(t)) = \sum_{j=1}^{n} w_{i,j}x_j(t) + \theta_i \tag{4}$$

The GHN can operate in different modes. If the evaluation of Eq. (3) is performed at all neurons at the same time, we say that the GHN is operating in *synchronous mode*. If the evaluation of Eq. (3) is performed at a single neuron each time, we say that the GHN is operating in *asynchronous mode*. Furthermore, we say that the GHN is operating in *randomly asynchronous mode* if the single activation neuron is selected from all n possible neurons of the GHN randomly with equiprobability.

A state $X = [x_1,x_2,\dots,x_n]^T$ of the GHN is called stable if

$$x_i = Sgn(H_i(X)) = Sgn\left(\sum_{j=1}^{n} w_{i,j}x_j + \theta_i\right) \quad (i = 1,2\dots,n) \tag{5}$$

i.e. if the state of the GHN never changes as a result of evolution in synchronous or asynchronous mode. Therefore, the definition of the stable state of the GHN is the same in any operation mode.

When the GHN operates from an initial state, it probably enters a stable state. It has been proved [4] that the GHN with nonnegative weights is stable in randomly asynchronous mode, and it is also shown by simulation experiments that almost any GHN having a stable state is stable in randomly asynchronous mode. Thus, the GHN can have the same stability as a Hopfield network, especially in randomly asynchronous mode, and therefore we can apply this neural network model to associative memory.

Obviously, the domain of attraction of a stable state dominates the behaviour of content-addressable memory of the GHN on this stable state. Thus, the concept of a domain of attraction is very important to associative memory. We now begin to study the domain of attraction of a stable state of the GHN. Since the domain of attraction of a stable state varies with the operation mode, we study it in synchronous and randomly asynchronous modes, respectively.

We now assume that $X^* = [x_1^*,x_2^*,\dots,x_n^*]^T$ is a stable state of the GHN, and study the domain of attraction of X^*.

Definition 1. Suppose the GHN operates in synchronous mode. We define the domain of attraction of X^* as

$$D(X^*) = \{X \in \{-1,1\}^n: \text{ X has the path of state transition to } X^*\} \tag{6}$$

When the GHN operates in synchronous mode and starts from an initial state $X \in D(X^*)$, it will evolve to X^* with certainty.

Definition 2. Suppose the GHN operates in randomly asynchronous mode. We define the domain of attraction of X^* as

$$D(X^*) = \{X \in \{-1,1\}^n : \text{ X has only possible paths of state transition to } X^*, \text{ but the other stable states or the random state cycles}\} \tag{7}$$

When the GHN operates in randomly asynchronous mode and starts from an initial state, its state will finally be attracted in a stable state, or a random state cycle defined to be a hole of the network in the state space in Ma [4]. A random state cycle of the GHN is a set of states within which the state of the GHN will always transit randomly when the state of the GHN becomes any one of the set of the states. Therefore, a random state cycle is closed and like a hole in the ground. However, there may exist a state cycle which is open to the stable states or random state cycles. Here if the GHN starts from an initial state $X \in D(X^*)$, there may exist an event where the state of the GHN will always transit in some open state cycle(s). But the probability of this event is zero, as the time converges to infinity, therefore the GHN will evolve to X^* with a probability of one.

Since the attractive action should be equal in all directions around X^* for associative memory, we further introduce the radius of the domain of attraction.

For clarity, we first define the t-neighbourhood of X^* as

$$N_t(X^*) = \{X \in \{-1,1\}^n : d_H(X,X^*) \le t\} \tag{8}$$

where $d_H(X,X^*)$ is the Hamming distance between X and X^*.

Definition 3. Suppose that $D(X^*)$ is the domain of attraction of X^*. If $N_t(X^*) \subset D(X^*)$, we say that $N_t(X^*)$ is a basin of attraction of X^*, and t is a possible radius of attraction of X^*. The radius of attraction of X^* is defined as the greatest of the possible radiuses of attraction of X^*.

Clearly, if we know $D(X^*)$, we certainly know the radius of attraction of X^*. However, $D(X^*)$ generally has a very complex structure, and there does not exist any effective method to directly compute it from $(\mathbf{W},\theta)$. Although we cannot obtain the radius of attraction of X^*, we will obtain a lower bound of it in the following. This lower bound will be called the absolute radius of attraction of X^*.

To study the radius of attraction of a stable state, we now introduce the dominating vector function as follows.

Definition 4. For a GHN $\mathbf{N} = (\mathbf{W},\theta)$, we define $E(X)$ as the dominating vector function of the network $\mathbf{N}$ at any state $X \in \{-1,1\}^n$ by

$$E(X) = [E_1(X),E_2(X),\ldots,E_n(X)]^{\mathrm{T}} \tag{9}$$

where

$$E_i(X) = \left(\sum_{j=1}^{n} w_{i,j}x_j + \theta_i\right)x_i \; (i = 1,2,\ldots,n) \tag{10}$$

By Eqs (3) and (5), we have

$$E_i(X^*) = \left(\sum_{j=1}^{n} w_{i,j}x^*_j + \theta_i\right)x^*_i \ge 0 \; (i = 1,2,\ldots,n) \tag{11}$$

for the stable state X^*. Obviously, we have the following theorem:

Theorem 1. For a GHN $\mathbf{N} = (\mathbf{W},\theta)$ and a state X, if the dominating vector function satisfies

$$E_i(X) > 0 \; (i = 1,2,\ldots,n) \tag{12}$$

then X is a stable state of the GHN.

Proof. When $E_i(X) > 0$, $H_i(X) = \sum_{j=1}^{n} w_{i,j}x_j + \theta_i$ and x_i have the same sign. Therefore,

$$x_i = Sgn(H_i(X)) \tag{13}$$

Because $E_i(X) > 0$ for $i = 1,2,\ldots,n$, Eq. (13) holds for $i = 1,2,\ldots,n$. Therefore, X is a stable state of the network.

By Theorem 1, we find that a state of the GHN is stable if the dominating vector components are all positive at this state. In fact this condition is not necessary, only in a special case where some of the dominating vector components become zero. Therefore, the dominating vector function dominates almost all the stable states of the GHN. Moreover, it also dominates the radii of attraction of the stable states, to a certain degree. We now discuss the relation between the radius of attraction of a stable state and the dominating vector components.

For the stable state X^*, we consider the dominating vector component $E_i(X^*)$. In the case of $x_i^* = 1$, $E_i(X^*) = \sum_{i=1}^{n} w_{i,j}x_j^* + \theta_i \geq 0$, there may be two kinds of $w_{i,j}x_j^*$ terms in the sum: the nonnegative and negative ones. We assume that an error appears in the jth component of X^* and the error pattern $X = [x_1^*,\ldots,x_{j-1}^*,-x_j^*,x_{j+1}^*,\ldots,x_n^*]^T$ is input to the GHN. If $w_{i,j}x_j^* < 0$, then $w_{i,j}x_j = -w_{i,j}x_j^* > 0$. Thus, the sum of $\sum_{j=1}^{n} w_{i,j}x_j + \theta_i > E_i(X^*)$, so the evolution of the network will always let the ith component of the state of the network be stable at x_i^* in synchronous or randomly asynchronous mode. Therefore, this kind of error is good for the attraction of the error pattern to X^* on the ith component. However, if $w_{i,j}x_j^* \geq 0$, the sum of $\sum_{j=0}^{n} w_{i,j}x_j + \theta_i$ will be decreased to $E_i(X^*) - 2w_{i,j}x_j^*$. If we expect that the ith component of the state of the network will also be stable at x_i^* after the evolution(s) of the network, it is sufficient that $E_i(X^*) - 2w_{i,j}x_j^* \geq 0$, i.e. $2w_{i,j}x_j^* \leq E_i(X^*)$. In this way, when there are more errors appearing in different components of X^*, the error pattern can also be attracted to X^* on the ith component if $E_i(X^*)$ is great enough. We define a nonnegative real vector $C = [c_1,c_2,\ldots,c_n]$ as

$$c_i = \begin{cases} w_{i,j}x_j^* & \text{if } w_{i,j}x_j^* > 0 \\ 0 & \text{if } w_{i,j}x_j^* \leq 0 \end{cases} \tag{14}$$

and rearrange its components to form another vector $C^* = [c_1^*,c_2^*,\ldots,c_n^*]$ in the following way:

$$c_1^* \geq c_2^* \geq \ldots \geq c_n^* \tag{15}$$

Then the maximum number of errors which can be corrected directly by $E_i(X^*)$ in the ith component – $r_i(X^*)$ – can be computed by

$$r_i(X^*) = \max\left\{ j : \sum_{k=1}^{j} 2c_k^* \leq E_i(X^*) \right\} \tag{16}$$

If $x_i^* = -1$, we only need to change C as

$$C_j = \begin{cases} -w_{i,j}x_j^* & \text{if } w_{i,j}x_j^* < 0 \\ 0 & \text{if } w_{i,j}x_j^* \geq 0 \end{cases} \tag{17}$$

We define the absolute radius of attraction of X – $r(X^*)$ – by

$$r(X^*) = \min\{r_1(X^*),r_2(X^*),\ldots,r_n(X^*)\} \tag{18}$$

From the above discussion and definitions, it can be easily verified that the GHN will evolve to X^* with certainty (probability one) when it starts from any $X \in N_{r(X^*)}(X^*)$ in synchronous (randomly asynchronous) mode. Thus, the absolute radius $r(X^*)$ is certainly a possible radius of attraction of X^*. Therefore, $r(X^*)$ is a lower bound of $R(X^*)$ – the radius of attraction of X^*. We further have a lower bound of the absolute radius of attraction of X^* by the following theorem.

Theorem 2. Suppose that X^* is a stable state of the GHN $\mathbf{N} = (\mathbf{W},\theta)$ and E is a positive real number. b is also a positive real number and satisfies

$$b \geq \max\{|w_{i,j}| : i,j = 1,2,\ldots,n\} \tag{19}$$

If $E_i(X^*) \geq E$ for $i = 1,2,\ldots,n$, then

$$r(X^*) \geq \lfloor E/2b \rfloor \tag{20}$$

where $\lfloor x \rfloor$ is the integer part of the real number x.

Proof. Because $b \geq |w_{i,j}| = |w_{i,j}x_j^*|$, then $b \geq c_i^*$ $(i = 1,2,\ldots,n)$:

$$r_i(X^*) = \max\left\{ j : \sum_{k=1}^{j} 2c_k^* \leq E_i(X^*) \right\}$$
$$\geq \max\left\{ j : \sum_{k=1}^{j} 2b \leq E \right\} = \lfloor E/2b \rfloor \tag{21}$$

for $i = 1,2,\ldots,n$. Hence, $r(X^*) \geq \lfloor E/2b \rfloor$.

3. The OPLA and OPLA Scheme

In this section, we propose the OPLA and OPLA scheme, and analyse the convergence of OPLA. We assume that the sample set $\mathcal{M} = \{X^1,X^2,\ldots,X^m\}$ is given. For a reasonable associative memory, and according to the above definitions and coding theory, the radius of attraction of X^k should be equal to or less than $h[k]$, which is computed by

$$h[k] = \lfloor (\min\{d_H(X^k,X^j) : j = 1,\ldots,k-1,k+1,\ldots,m\} - 1)/2 \rfloor \tag{22}$$

for $k = 1,2,\ldots,m$.

We further define h as the least one of $\{h[1],h[2],\ldots,h[m]\}$, i.e.

$$h = \min\{h[1],h[2],\ldots,h[m]\} \tag{23}$$

and refer to h as the maximum of possible uniform radii of attraction of the sample set.

We now construct the OPLA scheme by improving the perceptron learning algorithm [10] to OPLA. First, $t_i (0 \leq t_i \leq h[i])$ is selected as the object (required) value of the radius of attraction of the pattern $X^k (k = 1,2,\ldots,m)$, and b is the expected maximum of the absolute values of the weights. α is a positive constant as the learning rate, and we usually set is to be 0.1. δ is also a positive number, which is slightly greater than 0. For a set of the object values $(t_1,t_2,\ldots,t_m)$ of the radii of attraction of the sample patterns, the OPLA for the weights to and the threshold of neuron i is given as follows. Here $w_{i,i}$ is always set to be zero.

Step 1. Randomly select a set of initial weights $w_{i,1},\ldots,w_{i,i-1},w_{i,i+1},\ldots,w_{i,n}$, *as well as* θ_i *from the interval* $[-0.1,0.1]$.

Step 2. At time t, select a new sample pattern X^k from $[X^1,\ldots,X^m\}$ and use it to train the network.[2]

Step 3. Compute u_i by

$$u_i = Sgn\left(\sum_{j=1}^{n} w_{i,j}x_{k,j} + \theta_i - x_{k,i}(t_k b + \delta)\right) \tag{24}$$

Step 4. The weights and threshold are modified as follows. Let

$$\Delta w_{i,j} = \alpha(x_{k,i} - u_i)x_{k,j} \tag{25}$$

$$(j = 1,\ldots,i-1,i+1,\ldots,n);$$

$$\Delta\theta_i = \alpha(x_{k,i} - u_i) \tag{26}$$

Then for $j = 1,\ldots, i-1, i+1,\ldots,n$, if $|(w_{i,j} + \Delta w_{i,j})| \leq b, w_{i,j}$ is modified by $w_{i,j} + \Delta w_{i,j}$. Otherwise, $w_{i,j}$ remains unchanged. θ_i is always modified by $\theta_i + \Delta\theta_i$.

Step 5. If all weights $w_{i,j}$ and θ_i are unchanged for every sample pattern, then stop; otherwise let $t = t + 1$ and go to Step 2.

When the learning process has converged, we have the values of the weights to and the threshold of neuron i. Having completed these n learning processes of OPLA for i from 1 to n, we obtain a desired (W,θ), i.e. a desired GHN. We refer to this learning process as the *OPLA scheme* of object $\{t_1,t_2,\ldots,t_m\}$. If all object values $t_1,t_2,\ldots,t_m$ are equal to an integer t, we refer to it as the uniform OPLA scheme of object t. Moreover, the uniform OPLA scheme of object $t = h$ is referred to as the *uniform OPLA scheme* of maximum object.

According to Theorem 2, the trained GHN by the OPLA scheme enables each sample pattern X^k to be stable, with the radius of attraction being at least t_k. Therefore, the sample patterns really have reached their object values of the radii of attraction, respectively. Since the complexity of computation of OPLA is almost the same as that of the perceptron learning algorithm, we can thus implement OPLA as easily as the perceptron learning algorithm. As to the convergence of OPLA, it has been shown by a number of experiments that, if there exists such a GHN, (i.e. a set of (W,θ)) which satisfies the following inequalities

$$|w_{i,j}| \leq b;\ (i,j = 1,2,\ldots,n) \tag{27}$$

$$E_i(X^k) = \left(\sum_{j=1}^{n} w_{i,j}x_{k,j} + \theta_i\right) x_{k,i} \geq t_k\, b + \delta\ (i = 1,\ \ldots,\ n, k = 1,\ \ldots,\ m) \tag{28}$$

OPLA converges to a desired set of $\{w_{i,1},\cdots, w_{i,i-1}, w_{i,i+1},\cdots,w_{i,n},\theta_i\}$ for each $i(i = 1,2,\ldots,n)$ as the time t becomes large enough. We now try to analyse the reason for the convergence of OPLA in the remaining part of this section.

If $\{w_{i,1}, \ldots, w_{i,i-1}, w_{i,i+1}, \ldots, w_{i,n},\theta_i\}$ satisfies the above conditions (i.e. Eqs (27) and (28)), then there is a hyperplane

$$w_{i,1}x_1 + \ldots + w_{i,i-1}x_{i-1} + w_{i,i+1}x_{i+1} + \ldots + w_{i,n}x_n + \theta_i = 0 \tag{29}$$

which linearly separates $\{X^1(i),\ldots,X^m(i)\}$ according to $\{x_{i,1}, \ldots, x_{i,m}\}$, where

$$X^k(i) = [x_{k,1}, \ldots, x_{k,i-1}, x_{k,i+1}, \ldots, x_{k,n}]^T$$
for $k = 1, \ldots, m$

Furthermore, if we let

$$W_i = [w_{i,1}, \ldots, w_{i,i-1}\ w_{i,i+1}, \ldots, w_{i,n}]^T$$

then

$$\begin{cases} W_i^T X^k(i) + \theta_i \geq t_k b + \delta, & \text{if } x_{k,i} = 1 \\ W_i^T X^k(i) + \theta_i \leq -(t_k b + \delta), & \text{if } x_{k,i} = -1 \end{cases} \tag{30}$$

for $k = 1,2, \ldots, m$.

In comparison with perceptron learning algorithm, OPLA limits $|w_{i,j}| \leq b$ for $j = 1, \ldots, i-1, i+1, \ldots, n$ and make each $X^k(i)$ be in the correct side of the hyperplane with a proper distance, which is determined by the constraint $E_i(X^k) = (W_i^T X^k(i) + \theta_i)x_{k,i} \geq t_k b + \delta$. We now consider $E_i(X)$ as a function of $\mathbf{W}$ and θ, and observe the variation of these $E_i(X^k)(k = 1, \ldots, m)$ in the learning process. It is clear that if $E_i(X^1), \ldots, E_i(X^m)$ can reach or exceed their object values $t_1 b + \delta, \ldots,t_m b + \delta$, respectively, in a finite number of times, OPLA is convergent.

We consider the learning process of OPLA. For each time t, we introduce

$$\lambda(t) = [\lambda_1(t),\ldots,\lambda_n(t)]^T \tag{31}$$

where

$$\lambda_j(t) = \begin{cases} = 1, & \text{if } |w_{i,j}(t) + \Delta w_{i,j}| \leq b, j \neq i \\ = 0, & \text{if } |w_{i,j}(t) + \Delta w_{i,j}| > b, j \neq i \\ = 0, & \text{if } j = i \end{cases} \tag{32}$$

Since $w_{i,j}(0)$ is randomly selected in $[-0.1,\ 0.1]$ and b is much greater than 0.1, then $|w_{i,j}(0)| <<$

[2]Note that we can simply select X^k from X^1 to X^m repeatedly in the implementation.

b. Therefore, $\lambda_j(t)$ is 1 for $j = 1, \ldots, i-1, i+1, \ldots, n$ at the beginning stage of the learning process. On further learning, some $\lambda_j(t)$ may be zero for $j \neq i$, but it will quickly be changed to 1 in the sequential times, because $\Delta w_{i,j}$ varies in the positive and negative signs according to the selected samples. So we can assume that the great majority of $\lambda_j(t)$ are 1 in the whole learning process of OPLA. In other words, $n - 1 - \sum_{j=1}^{n} \lambda_j(t)$ is much smaller in comparison with n.

With these symbols, we have

$$w_{i,j}(t+1) = w_{i,j}(t) + \lambda_j(t)\Delta w_{ij} \tag{33}$$

If the sample pattern X^k is selected at time t and $u_i \neq x_{k,i}$, it is clear that $E_i(X^k)$ cannot reach the object value $t_k b + \delta$. After the modification of OPLA, we have

$$\Delta E_i(X^k) = \alpha\left(\sum_{j=1}^{n} \lambda_j(t) + 1\right)(x_{k,i} - u_i)x_{k,i} > 0 \tag{34}$$

Therefore, $E_i(X^k)$ has increased after this modification.

On the other hand, the sequential modification by the other sample $X^{k'}$ may cause a drawback for $E_i(X^k)$. In this case, we have

$$\Delta E_i(X^k) = \alpha\left(\sum_{j=1}^{n} \lambda_j(t)x_{k,j}x_{k',j} + 1\right)(x_{k',i} - u_i)x_{k,i} \tag{35}$$

If $\Delta E_i(X^k)$ is negative, a drawback is actually made. We now compare the absolute values of $\Delta E_i(X^{(k)})$ in two cases. In fact, the sample patterns $X^{k'}$ and X^k should have enough Hamming distance in order to be stored in a GHN. Then $\left|\sum_{j=1}^{n} x_{k,j}x_{k',j}\right|$ will become much smaller in comparison with n. Thus $\left|\sum_{j=1}^{n}\lambda_j(t)x_{k,j}x_{k',j} + 1\right|$ is also much smaller than $\sum_{j=1}^{n}\lambda_j(t) + 1$. Therefore, the decrease of $E_i(X^k)$ with $X^{k'}$ is much smaller than the increase of $E_i(X^k)$ with X^k.

By above analysis, $E_i(X^k)$ increases considerably when X^k is selected for the modification, but it may increase or decrease slightly when the other sample pattern $X^{k'}$ is selected for the modification. We see that the key mechanism of OPLA is that the modification of the weights and threshold enables $E_i(X^k)$ to increase on average in a period of m times to the object value. Since the object value of $E_i(X^k)$ is fixed to be $t_k b + \delta$, $E_i(X^k)$ will reach or exceed it in a finite number of times. So $E_i(X^k)$ will increase in m-time batches to the object value if $E_i(X^k)$ is lower than the object value. Furthermore, two or more $E_i(X^k)$ for different integers of k can increase to their object values in the same way synchronously. When some $E_i(X^k)$ first reaches or exceeds the object value, it may decrease slowly in sequential time. If it turns out to be lower than the object value again, it will exceed the object value quickly by the recent modification with X^k. Thus, when $E_i(X^k)$ reaches or exceeds the object value at some time, it will remain greater than the object value in almost all the times. Therefore, all $E_i(X^k)$ for $k = 1, \ldots, m$ will reach or exceed the object values, respectively, at a certain time. That is, OPLA will have converged to a desired set of weights and threshold by this time.

We have made an analysis on the convergence of OPLA when there exists a desired GHN under a set of object values. Although the analysis is not so strict as a mathematical proof, it is reasonable and heuristic. Furthermore, it is consistent with the empirical results.

4. The Simulation Results

In this section, several simulations are carried out to evaluate the performance of OPLA or the OPLA scheme. Our simulation experiments were undertaken on a sample set of ten Arabic numerals {0,1,2,3,4,5,6,7,8,9 } using the OPLA scheme and uniform OPLA scheme. Ten sample patterns are expressed by 7 $\times$ 7 pixies, as shown in Fig. 1. Based on the Hamming distances between these sample patterns, we have

$$(h[1],h[2],h[3],h[4],h[5],h[6],h[7],h[8],h[9],h[10]) = (3,6,5,4,9,4,5,8,3,4)$$

and $h = 3$. We take a GHN of 49 neurons and use this sample set to train it for associative memory. One simulation experiment consists of two processes. In the first process, OPLA is applied to train the network. In our experiments, b and δ of OPLA are always selected to be 100 and 1, respectively. When this learning process ends with success, i.e.

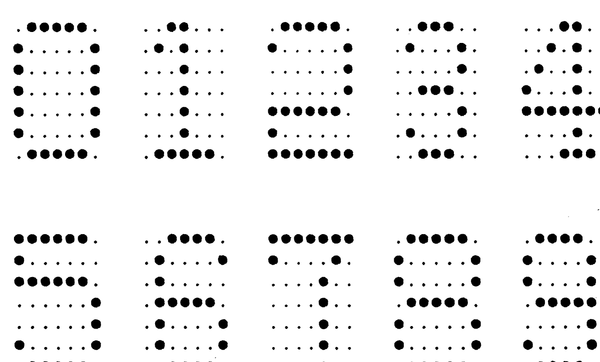

Fig. 1. Sample patterns of ten arabic numerals {0,1,2,3,4,5,6,7,8,9}.

a GHN is obtained, we turn to the second process by which the radii of attraction of the sample patterns are estimated as follows.

For the sample pattern $X^k (k = 1,2, \ldots, 10)$ and the number $j (j = 1,2, \ldots, 9)$ (Here 9 is the greatest of $\{h[1], h[2], \ldots, h[10]\}$), we randomly select 1000 initial patterns with a Hamming distance of j to X^k. These initial patterns can be considered as X^k polluted by j errors in some j components of the pattern. Then the trained network operates in randomly asynchronous mode with each initial pattern. We check whether the network finally evolve to X^k or not. If the network evolves to X^k for all 1000 polluted patterns, we are sure that j is a possible radius of attraction of X^k. In this way, we can estimate the radius of attraction of X^k.

We first carry out a simulation experiment with the uniform OPLA of object $t = 2$. The learning process of the OPLA scheme is completed successfully with a network, and the simulation results are listed in Table 1.

In Table 1 as well as two other tables in this paper, X^k in the first row represents the sample pattern, t_k in the second row represents the object value of the radius of attraction of X^k used in the OPLA scheme; $R(X^k)$ in the third row represents the estimated radius of attraction of X^k on the trained network; $R(X^k)/h[k]\%$ in the last row represents the percentage of achieving the maximum reasonable value of the radius of attraction of X^k – $h[k]$ by $R(X^k)$. According to Table 1, we can find that the trained network has the required function of associative memory. In fact, its function is even better than what we expect. The radii of attraction of five sample patterns are equal to 2 as we require, but the radii of attraction of the other five sample patterns is greater than 2. We see that the sample patterns may have different actual radii of attraction on the trained network, even if they have the same object value of the radius of attraction in the OPLA scheme. Moreover, the actual radius of attraction of a sample pattern X^k seems to have a relation with $h[k]$. Although we have obtained a desired network, the percentage of achieving the maximum reasonable value of radius of attraction is really low for some sample patterns. To improve these results, we try to increase the object value of the radius of attraction t or $\{t_1, t_2, \ldots, t_{10}\}$.

We then carry out a simulation experiment with the uniform OPLA of maximum object $t = h = 3$. The learning process cannot be completed successfully under this uniform object, but when the learning process is forced to stop after a large number of iterations, we have still obtained a useful network by which the radius of attraction of the first pattern 0 is 2 instead of the object value 3, and the radii of attraction of the other sample patterns are either equal to or greater than 3. Moreover, the radii of attraction of some sample patterns is greater than the corresponding radii of attraction listed in Table 1.

We further carry out a simulation experiment with the OPLA scheme of object $\{t_1 = 2, t_2 = 4, t_3 = 3, t_4 = 2, t_5 = 5, t_6 = 3, t_7 = 3, t_8 = 5, t_9 = 2, t_{10} = 3\}$. The learning process is completed successfully, and the simulation results are listed in Table 2.

From Table 2, we find that six radii of attraction are greatly increased, and the percentages $(R(X^k)/h[k])\%$ are improved to a satisfactory level. The radius of attraction of sample pattern 7 is even greater than $h[8] = 8$, which may be caused by the fact that some sample patterns cannot have their radii of attraction reach the corresponding object values. From this experiment, we see that the OPLA scheme with a set of carefully selected individual object values for the sample patterns is more valuable for associative memory than the uniform OPLA scheme. However, it is difficult to select an optimum set of these individual object values. One possible method is that we begin to let it be $\{h[1], \ldots, h[m]\}$ and test it using the OPLA scheme. If the learning process can be completed successfully, this set of object values is just the optimum. Otherwise, we slowly decrease the individual object values and test it using the OPLA scheme until the learning process is completed successfully. Then we have obtained the optimum set of object values as well as the network. In fact, $\{2,4,3,2,5,3,3,5,2,3\}$ is optimum for the sample patterns of ten Arabic numerals in our experiments.

In the above experiments, the trained GHN are operating in randomly asynchronous mode in the second process. We now let the trained network

Table 1. Simulation result of ten Arabic numerals with the uniform OPLA of object $t = 2$

X^k	0	1	2	3	4	5	6	7	8	9
t_k	2	2	2	2	2	2	2	2	2	2
$R(X^k)$	2	4	2	2	4	2	3	3	2	3
$(R(X^k)/h[k])\%$	67	67	40	50	44	50	60	38	67	75

Table 2. The simulation result of ten Arabic numerals with the OPLA scheme

X^k	0	1	2	3	4	5	6	7	8	9
t_k	2	4	3	2	5	3	3	5	2	3
$R(X^k)$	2	6	3	2	8	4	3	9	2	3
$(R(X^k)/h[k])\%$	67	100	75	50	89	100	75	113	67	75

operate in synchronous mode, and estimate the radii of attraction of the sample patterns in the same way. We carry out a simulation on the trained GHN of the above experiment to estimate the radii of attraction of the ten sample patterns in synchronous mode. The simulation results are listed in Table 3.

From the data listed in Table 3, we find that the radii of attraction of the ten sample patterns are really equal to or greater than the object values, respectively. Moreover, four radii of attraction are obviously improved, in comparison with the results of the randomly asynchronous mode listed in Table 2. Therefore, the OPLA scheme is also useful and effective for the GHNs in synchronous mode for associative memory.

In comparison with the other methods, we also carry out two simulation experiments on the ten numeral patterns to check the performance of the sum-of-outer product learning scheme with a Hopfield network and the weighted sum-of-outer product learning scheme with a GHN. In fact, it is shown by one simulation experiment that the ten sample patterns cannot all be stable on the Hopfield network constructed through the sum-of-outer product learning scheme. By another simulation experiment, the radii of attraction of the ten sample patterns are all 1 or 2 on the GHN trained through the weighted sum-of-outer product learning scheme [6]. As a result of the simulation experiments, the OPLA scheme with a GHN is more effective for associative memory than the sum-of-outer product learning scheme with a Hopfield network. It is even more effective than the weighted sum-of-outer learning scheme with an asymmetric Hopfield network.

Table 3. Simulation result of ten Arabic numerals in synchronous mode with the OPLA scheme

X^k	0	1	2	3	4	5	6	7	8	9
t_k	2	4	3	2	5	3	3	5	2	3
$R(X^k)$	2	8	6	2	8	5	5	9	2	5
$(R(X^k)/h[k])$%	67	133	120	50	89	80	100	113	67	120

5. Conclusion

In this paper, we have analysed the radius of attraction of a stable state, and proposed the OPLA and OPLA scheme on generalised Hopfield networks for associative memory. By introducing the dominating vector function, the absolute radius of attraction is defined and a lower bound of it has been obtained. By improving the perceptron learning algorithm to OPLA, the OPLA scheme is constructed to store each sample pattern, with its radius of attraction being equal to or greater than an object value. A heuristic analysis is made on the convergence of OPLA when there exists a desired GHN under a set of object values. The OPLA can be implemented as easily as the perceptron learning algorithm, and it has been shown by the simulation experiments that the OPLA scheme is effective for associative memory.

References

1. Hopfield JJ. Neural networks and physical system with emergent collective computational abilities. Proc Nat Academy Sci USA 1982; 79: 2554–2558
2. McEliece RE et al. The capacity of the Hopfield associative memory. IEEE Trans Inform Theory 1987; IT-33: 461–483
3. Amari S, Maginu K. Statistical neurodynamics of associative memory. Neural Networks 1988; 1: 63–73
4. Ma J. The stability of the generalised Hopfield networks in randomly asynchronous mode. Neural Network 1997; 6: 1109–1116
5. Gardner E. The space of interactions in neural network models. J Physics A: Mathematics and General 1988; 21: 257–270
6. Abbott LF, Kepler TB. Optimal learning in neural network memories. J Physics A: Mathematics and General 1989; 22: L711–L717
7. Wang T et al. Learning algorithm in unequally-weighted associative memory. Pattern Recognition and Artificial Intelligence (in Chinese) 1993; 7: 43–48
8. Venkatesh SS, Psaltis D. Linear and logarithmic capacities in associative memory. IEEE Trans Inform Theory 1989; IT-35: 558–568
9. Ma J. The asymmetric Hopfield model for associative memory. Proc Int Joint Conf Neural Networks 1993; 3: 2611–2614
10. Rosenblatt F. Principles of Neurodynamics. Spartan Books, New York, 1962