Pergamon

Neural Networks, Vol. 10, No. 6, pp. 1109–1116, 1997

Printed in Great Britain
0893–6080/97 $17.00+.00

PII: S0893-6080(97)00026-9

CONTRIBUTED ARTICLE

The Stability of the Generalized Hopfield Networks in Randomly Asynchronous Mode

JINWEN MA

Institute of Mathematics, Shantou University

(*Received* 28 *March* 1995; *accepted* 10 *February* 1997)

Abstract—*We present a study on the stability of the generalized Hopfield network in randomly asynchronous mode. First, the stability is investigated from the state space of the network. We introduce a concept of hole and define two kinds of stability in randomly asynchronous mode. By mathematical inductive method, we have proved that a generalized Hopfield network with non negative weights is strictly stable, that is, the network evolves to a simple hole—stable state with probability one when it starts at any initial state. For the general case, we made the simulation experiments on the networks with the number of neurons from 5 to 10. The empirical results have shown that almost any generalized Hopfield network with simple holes is strictly stable. © 1997 Elsevier Science Ltd.*

Keywords—Neural network, Hopfield network, Stability, Associative memory.

1. INTRODUCTION

The discrete-time Hopfield network was proposed mainly as an associative memory model (Hopfield, 1982). It is a dynamical system uniquely defined by (W,θ) where W is a symmetric zero-diagonal real weight matrix, and θ is a real threshold vector. By introducing a special energy function, Hopfield (1982) proved that the network in an asynchronous mode (randomly or deterministically) will evolve to a stable state with any initial state, which is the key to associative memory.

In this kind of neural network, when the weight matrix is asymmetric and zero-diagonal, we usually call it an asymmetric (discrete-time) Hopfield network. In this paper, we define a generalized Hopfield network as such a kind of network with a general (asymmetric or symmetric) zero-diagonal real weight matrix. In recent years, for the purpose of associative memory, several learning algorithms or schemes have been established on the generalized Hopfield networks (refer to, e.g., Gardner, 1988; Abbott, 1989; Venkatesh & Psaltis, 1989; Allen & Alspector, 1990; Xu & Tsai, 1990; Ma, 1993; 1995). In fact, these schemes are all based on making a set of given patterns to be the stable states of a network. Many experiments show that the effect of associative memory with the network operating asynchronously is better and more reasonable than that with the network operating synchronously, since the state of a synchronous network has more probability to be trapped in a limit cycle than to converge to a stable state. But the stability of the generalized Hopfield network in any kind of asynchronous mode has not been investigated in depth.

In this paper, we present a study on the stability of the generalized Hopfield networks in randomly asynchronous mode. Hereafter we refer to Generalized Hopfield Network as GHN for short. In Section 2, we investigate the pair-wised network—a more general form of GHN—from its state space and make it be equivalent to a directed hypercube. Then we generalize the stable state to the hole and prove that the network in randomly asynchronous mode will drop in a hole with probability one when it starts at any initial state. In Section 3, we turn to discuss GHNs and prove that the GHN with non-negative weights is strictly stable. For the general case, we make the simulation experiments on the networks with the number of neurons from 5 to 10 and the empirical results show that almost any GHN having simple holes is strictly stable. Our conclusions appear in Section 4.

2. THE STABILITY OF THE PAIR-WISED CONNECTED NEURAL NETWORKS

We first consider the pair-wised connected neural network of n neurons which is a more general form of

Acknowledgements: This paper was supported by China Tianyuan Funds under Project 1937015.

Requests for reprints should be sent to Jinwen Ma, Institute of Mathematics, Shantou University, Shantou, 515603, Guangdong, P.R.C.; Tel: 86 754 8510000 Ext 32170; Fax: 86 754 8510505.

GHN or discrete-time Hopfield network. Here a pairwised connected neural network is defined to be a single layer consisting of n neurons which are connected to one another. Every neuron is in one of two possible states, either 0 or 1 (when the two possible states are -1 and 1 instead of 0 and 1, it can be easily verified that the network has the same properties of stability discussed in this paper). The state of neuron i at time t is denoted by $x_i(t)$. The state of the neural network at time t is the vector $X(t) = [x_1(t),x_2(t),...,x_n(t)]^T$. The network can operate in different modes. If the computation is performed at all neurons at the same time, we say that the network is operating in synchronous mode. If the computation is performed at a single neuron at each time, we say that the neural network is operating in asynchronous mode. Furthermore, we say that the network is operating in a randomly asynchronous mode if the single activation neuron is randomly selected subject to one probability distribution. And we say that the network is operating in a deterministically asynchronous mode if the single activation neuron is deterministically selected by a certain rule (function).

In any asynchronous mode, the label of the selected neuron forms a sequence as the network evolves. Such sequence will be called an asynchronous sequence for the evolution of the network. In fact, the characteristic of the asynchronous sequence and the structure of the network together determine the stability of the network in an asynchronous mode.

In this paper, we study the stability of the network in randomly asynchronous mode. For clarity, we only consider a typical randomly asynchronous mode which is accurately described as follows.

The network starts with an initial state. Then at each time, a neuron is randomly selected from all possible n neurons of the network with equiprobability. If the selected neuron is neuron i and the time is $t + 1$, then the state of neuron i is computed from the state $X(t)$ of the network at time t by

$$x_i(t+1) = F_i(X(t)) = F_i(x_1(t), ..., x_{i-1}(t), x_{i+1}(t), ..., x_n(t)) \quad (1)$$

where F_i is the activation function of neuron i from $\{0,1\}^n$ to $\{0,1\}$ (in fact, it only depends on the $(n-1)$ components of $X(t)$ except $x_i(t)$). The states of other neurons remain unchanged, i.e.

$$x_j(t+1) = x_j(t) \quad (j \neq i) \quad (2)$$

On the other hand, if $X(t+1)$ is computed from the current state $X(t)$ by performing the evaluation eqn (1) at every neuron of the network, the mode of operation is synchronous.

A state $X = [x_1,x_2,...,x_n]^T$ of the network is called stable if

$$X_i = F_i(X) \quad (i = 1, 2, ..., n) \quad (3)$$

i.e. if the state of the network will never change as a result of evolution in any asynchronous or synchronous mode. Thus the stable state of the network in any operation mode is the same. However, the stability of the network in different operation mode may be different. In the following, we only discuss the stability of the network in randomly asynchronous mode.

We now observe the operating of the network from the state space $V^{(n)}(0,1) = \{0,1\}^n$ which consists of all possible 2^n vertices of n-dim unit hypercube. Every state of the network is considered as a vertex of $V^{(n)}(0,1)$ or a binary code of n bits. In an n-dim hypercube, each vertex has n edges connecting to its n neighbor vertices. Two neighbor vertices expressed as two n-dim binary vectors have one Hamming distance between them. We consider that the state (vertex) of the network is $X = [x_1,x_2,...,x_n]^T$ and neuron i is selected. According to the state evolution rule of eqns (1) and (2), if $F_i(X) \neq x_i$, i.e., $F_i(X) = 1 - x_i$, X will be transited to $[x_1,...,x_{i-1},1 - x_i,x_{i+1},...,x_n]^T$ which is defined as the ith neighbor vertex of X. We give a direction from X to its ith neighbor vertex on the edge between them. Else if $F_i(X) = x_i$, the state of the network X will not change at time $t + 1$. In this case, by eqn (1), $F_i([x_1,...,x_{i-1},1 - x_i,x_{i+1},...,x_n]^T) = F_i([x_1,...,x_{i-1},x_{i+1},...,x_n]) = F_i(X) = x_i$. Therefore when the ith neighbor vertex is the state of the network and neuron i is selected, the ith neighbor vertex of X, i.e., $[x_1,...,x_{i-1},1 - x_i,x_{i+1},...,x_n]^T$ will be transited to X. Then we give a direction from the ith neighbor vertex to X on the edge between them. In this way, we can give a direction on every edge of the hypercube. Thus the evolution function of the network defines a directed hypercube in the state space. On the contrary, a directed hypercube also defines a unique evolution function of the network. Therefore a network is functionally equivalent to a directed hypercube and we can study the network by its directed hypercube.

For the stability of the network, we introduce a concept of the hole in a directed graph.

DEFINITION 1. *Assuming that* $\mathbf{G}$ *is a directed graph and* $\mathbf{H}$ *is a subgraph of* $\mathbf{G}$. $\mathbf{H}$ *is called a hole of* $\mathbf{G}$ *if it satisfies:*

1. $\mathbf{H}$ *is a connected graph, i.e., any vertex of* $\mathbf{H}$ *can connect to any one of the other vertices of* $\mathbf{H}$ *by a path in* $\mathbf{H}$.
2. Any vertex outside $\mathbf{H}$ *can only connect to the vertices of* $\mathbf{H}$ *if it is possible. In other words, each vertex in* $\mathbf{H}$ *cannot connect to any vertex outside* $\mathbf{H}$.

For a network, if $\mathbf{H}$ is a hole of its directed hypercube, we say that $\mathbf{H}$ is a hole of the network. We also say that this set of vertices or states in $\mathbf{H}$ is a hole of the network.

The stable state is certainly a hole of the network. So the hole is just a generalization of the stable state. A hole may be simple, just like a stable state, but it may be complex and very large, even containing all the vertices. We further define a hole to be simple or complex as follows. A hole is simple if it contains only one vertex,

i.e., it is a stable state. A hole is complex if it contains two or more vertices.

The structure of a hole may be complicated. Figure 1 gives the structure sketches of three kinds of holes. In Figure 1, a small black circle represents a vertex (state). Figure 1a describes the structure of a simple hole—a stable state. Figure 1b describes the structure of a special complex hole—the limit cycle of the network in randomly asynchronous mode. The other edges connecting to the vertices of the hole are all omitted in the figure. In this kind of limit cycles, any two neighbor vertices have one hamming distance between them. Figure 1c describes an example of the general structure of a complex hole. It can be thought as a group of the cycles among which any cycle intersect or is tangent with one of the other cycles. The other edges connecting to the vertices of the hole are also omitted in the figure.

THEOREM 1. *Any network in randomly asynchronous mode defined by eqn (1) will drop in a hole of the network with probability one when it starts at any initial state in the state space.*

Proof. For any network defined by eqn (1), we consider its directed hypercube and divide all vertices into two sets $\mathcal{A}$ and $\mathcal{B}$. If a vertex (state) has a path connecting to a hole of the network, which means that the network has a positive probability to drop in the hole with this state as an initial state of the network, the vertex belongs to $\mathcal{A}$. Else if a vertex cannot connect to any hole of the network, the vertex belongs to $\mathcal{B}$. Now we prove that $\mathcal{B}$ is empty.

We suppose that $\mathcal{B}$ isn't empty. Then we say $X_1 \in \mathcal{B}$. Because X_1 cannot be a stable state, X_1 must connect to another vertex $X_2 \in \mathcal{B}$. (The vertex of $\mathcal{B}$ cannot connect to the vertices of $\mathcal{A}$ under the classification.) In this way on i^{th} step, $X_i \in \mathcal{B}$ may be in two cases. First, X_i is still different from any one of $X_1,X_2,...,X_{i-1}$. Second, X_i is equal to one of $X_1,X_2,...,X_{i-1}$. We say $X_i = X_k (k < i)$, then $X_k,X_{k+1},...,X_i$ form a cycle in the directed hypercube—the state space. Because this cycle, or even it with the precedently formed cycles which intersect or are tangent with it, cannot be a hole, there must be a new vertex in $\mathcal{B}$ which is connected from the cycle or cycles. We change X_i to be the new vertex in this case. Therefore, a new and different vertex can always be produced and added to the vertex sequence $X_1,X_2,...,X_i$ on each step. If we form the sequence unlimitedly, we certainly get infinite different vertices in $\mathcal{B}$. This is a contradiction with the finite number of the state space of the network. Therefore, $\mathcal{B}$ is empty.

As $\mathcal{B}$ is empty, $\mathcal{A}$ is the state space. Therefore, there must exist at least one hole in the directed hypercube and any vertex can connect to one of the holes. Then for each state (vertex), it certainly has a path to a hole. Since the path consists of a finite number of steps (edges) in the directed hypercube, the network can evolve to the hole in a finite number of steps when the asynchronous sequence is properly selected. Because the number of all the states of the network is finite, there certainly exists a finite number m such at the network will probably evolve to a hole in m steps from any initial state in $\{0,1\}^n$. Thus, by the assumption of the randomly asynchronous mode and over the classical probability space $\{1,2,...,n\}^m$ of the m-step asynchronous sequence, the network has a positive probability $P(X)$ to evolve to a hole from X (as an initial stage) in m steps. If X is already in a hole, $P(X) = 1$. We let p be the minimum of $P(X)$ over $\{0,1\}^n$. Then $p > 0$.

Now we consider the network evolves with an initial state X^0. The asynchronous sequence is generated in the randomly asynchronous mode. We express it in m-step block as follows:

$$S = (s_{1,1}, s_{1,2}, ..., s_{1,m}, s_{2,1}, s_{2,2}..., s_{2,m}, ..., s_{k,1}, s_{k,2}, ..., s_{k,m}, ...) \quad (4)$$

In the first block of m steps, the network starts with X^0. Then the network has the probability $P(X^0)$ to evolve to a hole. At the end of the first block, the state of the network is $X(m)$. We denote $X^1 = X(m)$. If X^1 is in a hole, then the network will always be in this hole. Otherwise, if X^1 is not in a hole, we need the second block to further make the network evolve to a hole. In this case, $1 - P(X^0)$ is the

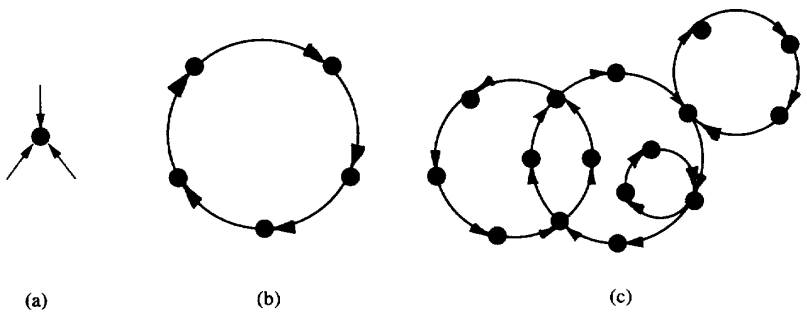


FIGURE 1. The structure sketches of the three kinds of holes of GHNs.

probability of the event E_1 that the network does not drop in a hole after the first block, i.e. X^1 is not in a hole.

In this way, at the end of the kth block, the state of the network is $X(km)$. We denote X^k is in a hole, the network will always be in this hole. Otherwise, if X^k is not in a hole. we still need the further block to make the network evolve to a hole. In this case, over the product probability space of k independent blocks, the probability of the event E_k that the network does not drop in a hole after the k blocks is

$$P(E_k) = (1 - P(X^0)(1 - P(X^1)...(1 - P(X_{k-1})) \leq (1-p)^k. \tag{5}$$

Because $0 < 1 - p < 1$, then $P(E_k)$ converges to zero as k tends to infinity. Thus the network with X^0 will finally evolve to a hole with probability one as the time tends to infinity. Therefore, the network will drop in a hole of the network with probability one when it starts at any initial state in the state space. The proof is completed.

By Theorem 1, we can consider each network defined by eqn (1) is stable with probability one in light of converging to the holes of it. In fact, besides the holes of the network, there may be some cycles in the directed hypercube. But they have the paths to some holes. Therefore, these cycles do not trap the state of the network with probability one as the time continues to infinity. In other words, these cycles trap the state of the network with probability zero. Theoretically, the network may always transit in these cycles when a very special asynchronous sequence appears. However, this case appears with probability zero. We can be sure that it does not appear in the application. By the empirical results, we always find the network quickly drops in a hole.

For associative memory, it is reasonable that the network converges to a stable state with any initial state in randomly asynchronous mode. According to Theorem 1, this is equivalent to that each hole of the network is simple. We define that these kind of networks are strictly stable. Hopfield network is an example of this kind of network. Furthermore, the following corollary is deduced from Theorem 1.

COROLLARY 1. *If the directed hypercube of a such network satisfies that every vertex has a path to a simple hole—stable state, the network is strictly stable.*

Proof. By Theorem 1, the network has holes. We suppose that there exists a complex hole **H** of the network or its directed hypercube. Then any vertex in **H** cannot connect to the vertices outside **H** according to the definition of the hole. So every vertex in **H** cannot have a path to a simple hole in the directed hypercube. This is contradictory with the fact that every vertex has a path to a simple hole. Therefore the network has only simple holes. Hence the network is strictly stable. The proof is completed.

A GHN may have complex holes. For associative memory, we want to know when a GHN is strictly stable and whether there are abundant GHNs with simple holes. Furthermore, how much is the probability (frequency) of the event that a GHN with simple holes is strictly stable? These problems are closely related to the associative memory of GHNs. In Section 3, we will discuss these problems on GHNs.

3. THE STABILITY OF THE GENERALIZED HOPFIELD NETWORKS (GHNS)

We now consider a GHN $\mathbf{N} = (W,\theta)$. Here W is an $n \times n$ zero-diagonal matrix with its element $w_{i,j}$ denoting the weight of the connection from neuron j to neuron i; θ is a vector of dimension n with its component θ_i denoting the threshold value of neuron i. The activation function $F_i(X)$ is expressed as follows:

$$F_i(X) = I(H_i(X)) = \begin{cases} 1 & \text{if } H_i(X) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where

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

Since $w_{i,i} = 0$, $H_i(X)$ is unconcerned with x_i, and so is $F_i(X)$ here.

When the weight matrix is symmetric, the network is a Hopfield network. As Hopfield (1982) proved by the energy function, a Hopfield network with any initial state will evolve to a stable state—a simple hole in any asynchronous mode. Then a Hopfield network has no complex hole. Thus a Hopfield network is strictly stable.

3.1. The Strict Stability of the GHNS with Nonnegative Weights

We now give another kind of strictly stable networks.

THEOREM 2. *A GHN with nonnegative weights is strictly stable.*

Proof. We prove this theorem by the mathematical inductive method. We begin with $n = 2$.[1] We define $\mathbf{N}^{(2)} = (W^2,\theta^2)$ to be a GHN of two neurons with nonnegative weights. The weight matrix and threshold vector are given as follows:

$$W^{(2)} = \begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix}, \quad \theta^{(2)} = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \tag{7}$$

where $a \geq 0$, $b \geq 0$. Then $H_1(X) = ax_2 + \theta_1$, $H_2(X) = bx_1 + \theta_2$.

[1] $n = 1$, the theorem certainly holds, but the network has no weight. For intuition with the condition of nonnegative weights of the network, here we begin with $n = 2$.

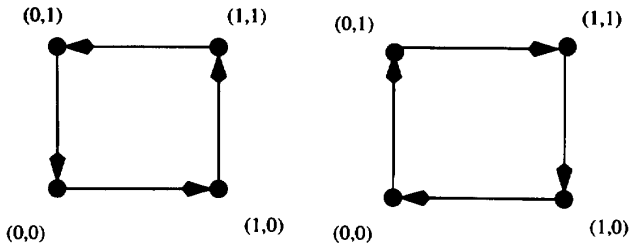


FIGURE 2. Two directed hypercubes of GHNs of two neurons with the complex holes.

The state space of $\mathbf{N}^{(2)}$ is $V^{(2)}(0,1) = \{(0,0),(0,1),(1,0),(1,1)\}$. From the directed hypercubes, we can easily verify that only two possible networks of two neurons are not strictly stable. Figure 2 gives the directed hypercubes of these two networks.

If a network of two neurons has the directed hypercube of Figure 2a, it should obey the transition rules in Table 1.

If a $\mathbf{N}^{(2)}$ obeys the transition rules of Table 1, the parameters of the network satisfy the following group of inequalities:

$$\begin{cases} \theta_1 & \geq 0 \\ \theta_2 & \geq 0 \\ a+\theta_1 & \geq 0 \\ b+\theta_2 & \geq 0 \end{cases} \tag{8}$$

Then we get that $a < -\theta_1$, and $\theta_1 \geq 0$, i.e., $a < 0$. This is a contradiction with $a \geq 0$. Therefore any $\mathbf{N}^{(2)}$ cannot obey the transition rules or have such a directed hypercube. In the same way, we can prove that any $\mathbf{N}^{(2)}$ cannot have the directed hypercube of Figure 2b. Then any $\mathbf{N}^{(2)}$ cannot have each of the two directed hypercubes. Thus any $\mathbf{N}^{(2)}$ is strictly stable, i.e., a GHN of two neurons with nonnegative weights is strictly stable.

Then we suppose that a GHN of $\mathbf{n}$ neurons with nonnegative weights is strictly stable. We define $\mathbf{N}^{(n)} = (W^{(n)},\theta^{(n)})$ to be a GHN of n neurons with nonnegative weights.

We now consider a GHN of $(n+1)$ neurons with nonnegative weights and define $\mathbf{N}^{(n+1)} = (W^{(n+1)},\theta^{(n+1)})$ as a GHN of $(n+1)$ neurons with nonnegative weights.

With $(n+1)$ neurons, the weight matrix and threshold vector are given as follows:

$$W^{(n+1)} = \begin{pmatrix} 0 & w_{1,2} & \cdots & w_{1,n} & w_{1,n+1} \\ w_{2,1} & 0 & \cdots & w_{2,n} & w_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & 0 & w_{n,n+1} \\ w_{n+1,1} & w_{n+1,2} & \cdots & w_{n+1,n} & 0 \end{pmatrix},$$

$$\theta^{(n+1)} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \\ \theta_{n+1} \end{pmatrix} \tag{9}$$

By Corollary 1, if any vertex has a path to a simple hole in the directed hypercube, the network is strictly stable. In the pointview of state transition, if the network starts with an initial state X and finally evolves to a stable state X^* under any asynchronous sequence, there is certainly a path from X (as a vertex) to X^* (as a simple hole) in the directed hypercube. Therefore in order to prove the strict stability of $\mathbf{N}^{(n+1)}$, we need only to prove that there always exists an asynchronous sequence for the network to evolve to a stable state from any initial site. In the following, we construct such an asynchronous sequence for any initial state by the induction hypothesis and the properties of $\mathbf{N}^{(n+1)}$.

Suppose that $X = [x_1,x_2,\ldots,x_n,x_{n+1}]^T \in V^{(n+1)}(0,1)$ is an initial state of $\mathbf{N}^{(n+1)}$, i.e., $X(0) = X$. Let $X(t) =$

TABLE 1

with $(W^{(n)},\theta^{(n)})$ as follows.

$$W^{(n)}=\begin{pmatrix} 0 & w_{1,2} & \cdots & w_{1,n-1} & w_{1,n} \\ w_{2,1} & 0 & \cdots & w_{2,n-1} & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{n-1,1} & w_{n-1,2} & \cdots & 0 & w_{n-1,n} \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n-1} & 0 \end{pmatrix},$$

$$\theta^{(n)}=\begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{pmatrix} \tag{10}$$

According to the induction hypothesis, $\mathbf{N}^{(n)}$ is strictly stable. By the proof of Theorem 1, there is a path in the directed hypercube of $N^{(n)}$ from $[x_1,x_2,...,x_n]^T = X^{(n)}$ to a stable state of $\mathbf{N}^{(n)}$, say $X' = [x'_1,x'_2,...,x'_n]^T$. Then there is an asynchronous sequence S_1 by which the network of $\mathbf{N}^{(n)}$ can evolve to X' from $X^{(n)}$. With S_1, the state vector of $N^{(n+1)}$ is $[x'_1,x'_2,..,x'_n,0]^T$ at the end.

We then select neuron $(n+1)$ to perform the evaluation. After the evolution, if $x_{n+1}(t)$ remains unchanged, $[x'_1,x'_2,..,x'_n,0]^T$ is certainly a stable state of $\mathbf{N}^{(n+1)}$. Thus $\mathbf{N}^{(n+1)}$ has already evolved to a stable state with the initial state X by the asynchronous sequence $S = S_1$.

On the contrary, if $x_{n+1}(t)$ has changed to 1, the state of the network becomes $X(t) = [x'_1,x'_2,..,x'_n,1]^T$. Then we obtain the following inequality.

$$w_{n+1,1}x'_1 + w_{n+1,2}x'_2 + ... + w_{n+1,n}x'_n + \theta_{n+1} \geq 0 \tag{11}$$

We begin the second stage of selecting the performing neuron again only from the front n neurons in the evolution steps. At this stage, $\mathbf{N}^{(n+1)}$ is functionally degenerated to another $\mathbf{N}^{(n)}$ of the front n neurons with $(W^{(n)},\theta^{(n)})$ as follows:

$$W^{(n)}=\begin{pmatrix} 0 & w_{1,2} & \cdots & w_{1,n-1} & w_{1,n} \\ w_{2,1} & 0 & \cdots & w_{2,n-1} & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{n-1,1} & w_{n-1,2} & \cdots & 0 & w_{n-1,n} \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n-1} & 0 \end{pmatrix},$$

$$\theta^{(n)}=\begin{pmatrix} \theta_1 + w_{1,n+1} \\ \theta_2 + w_{2,n+1} \\ \vdots \\ 0_n + w_{n,n+1} \end{pmatrix} \tag{12}$$

For this $\mathbf{N}^{(n)}$, we have the proposition that the state variable $x_i(t)$ of neuron i (for $i = 1,2,...,n$) with the initial state X' will not decrease in the following steps. It can be verified as follows.

We begin with the first evolution step and assume that neuron i is selected to perform the evaluation. When $x'_i = 1$, since X' is the stable state of the first degenerated network described by eqn (10), we have

$$w_{i,1}x'_1 + w_{i,2}x'_2 + ... + w_{i,n}x'_n + \theta_{i+1} \geq 0. \tag{13}$$

Because $w_{i,n+1} \geq 0$, we certainly have

$$w_{i,1}x'_1 + w_{i,2}x'_2 + ... + w_{i,n}x'_n + w_{i,n+1} + \theta_i \geq 0. \tag{14}$$

So that $x_i(t)$ will be 1 after the step. When $x'_i = 0$,x_i(t) can only be 0 or 1. Then $x_i(t)$ cannot be decreased. Therefore the state variable of each neuron of the network doesn't decrease in the first step.

At the beginning of the second step, the state vector of the network may be in two situations. Firstly it is X' (unchanged). By the analyses of evolution of the first step, we know that the state variable of each neuron will not decrease in the second step. Secondly it is changed to $[x'_1,...x'_{i-1},1,x'_{i+1},...,x'_n]^T$ instead of $X' = [x'_1,...,x'_{i-1},0,x'_{i+1},...,x'_n]^T$. Then the state vector has an additional 1 component in comparison with x'. Because $w_{j,i} \geq 0$, $H_j(X(t))$ will be increased or unchanged in the second step. Therefore the state variable of each neuron of the network will not decrease in the second step. By the above analyses of two situations, we can obtain that the state variable of each neuron of the network doesn't decrease in the second step.

In the same way, the state of each neuron of the network doesn't decrease in the following steps. Summing up the above discussion, we have verified the proposition.

From the induction hypothesis, this $\mathbf{N}^{(n)}$ is still strictly stable. So there is an asynchronous sequence S_2 by which the $N^{(n)}$ evolves from X' to a stable state, say $X'' = [x''_1,x''_2,...,x''_n]^T$.

Now we return to $\mathbf{N}^{(n+1)}$ with the state $[x''_1,x''_2,...,x''_n,1]^T$. We select neuron $(n+1)$ for the second time. By the proposition above, we certainly have

$$x''_i \geq x'(i = 1,2,...,n). \tag{15}$$

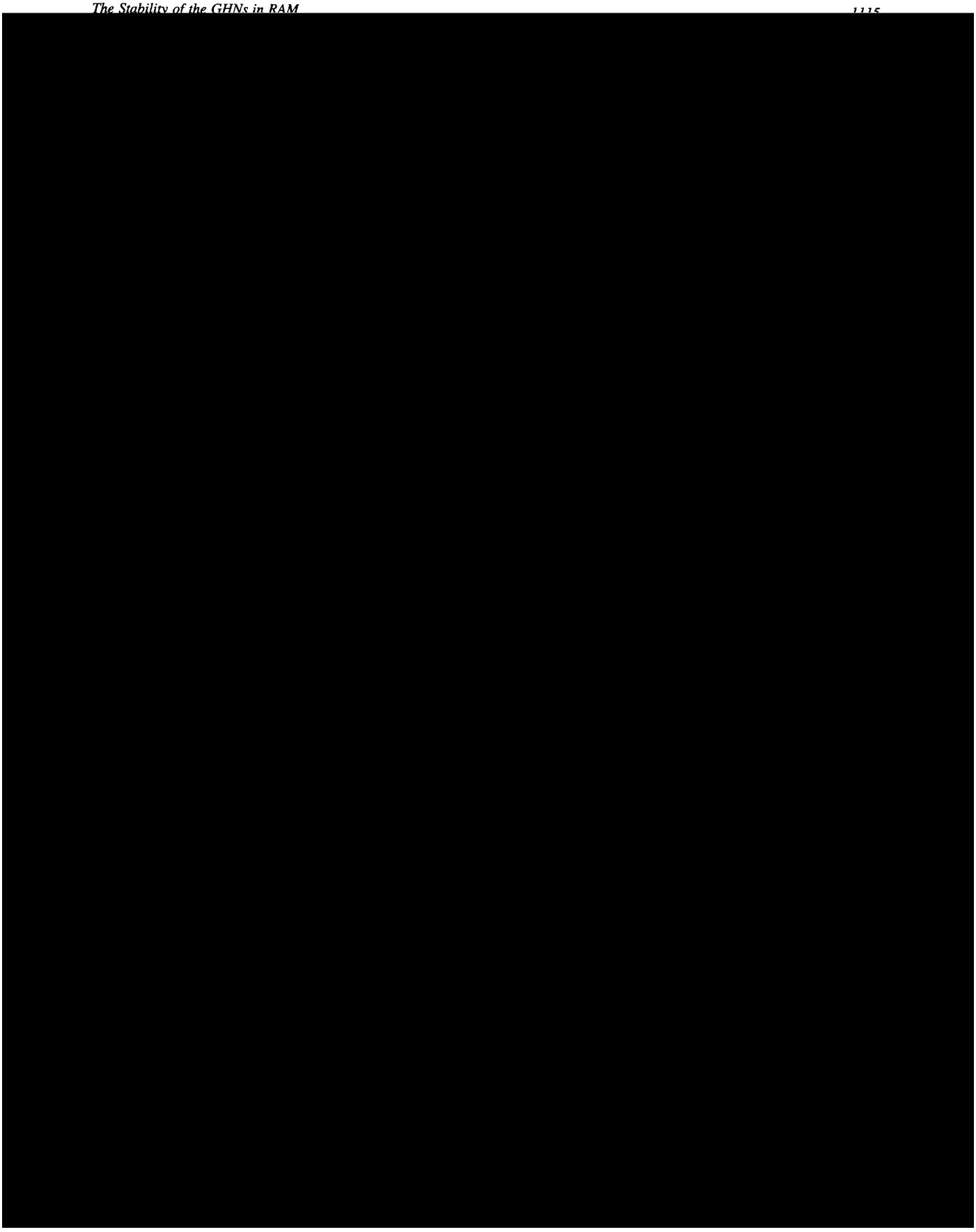
By eqns (11) and (15), we have

$$w_{n+1,1}x''_1 + w_{n+1,2}x''_2 + ... + w_{n+1,n}x''_n + \theta_{n+1} \geq 0, \tag{16}$$

that is, the state of neuron $(n+1)$ is still 1. Therefore the state $[x''_1,x''_2,...,x''_n,1]^T$ is a stable state of $\mathbf{N}^{(n+1)}$. Thus the $N^{(n+1)}$ has evolved to a stable state with the initial state X by the asynchronous sequence $S = (S_1,n+1,S_2)$.

Summing up the above results, there always exists an asynchronous sequence for $\mathbf{N}^{(n+1)}$ to evolve to a stable state from any initial state X with $x_{n+1} = 0$.

(2) $x_{n+1} = 1$. In the same way, we select the performing neuron only from the front n neurons in the evolution steps at the first stage. In this case, as we fix the state of neuron $(n+1)$, $\mathbf{N}^{(n+1)}$ is again degenerated to the $\mathbf{N}^{(n)}$ of



asynchronous mode. The acting operation mode is generally used as a practicable and reasonable randomly asynchronous mode. However, according to the proof of Theorem 1, we can easily find that the theoretical results of the stability obtained in this paper are also true for any randomly asynchronous mode subject to a probability distribution by which each neuron is selected with a positive probability. These results are significant to associative memory. Moreover, we made some other simulation experiments in synchronous or some deterministically asynchronous modes. In this case, we find that operating in a deterministically asynchronous mode or synchronous mode, a GHN may be attracted in a stable or state cycle, even when it is strictly stable. Most of states in the state space (as initial states) will be trapped in the cycles instead of stable states.

4. CONCLUSION

We have begun to explore the stability of GHNs in randomly asynchronous mode. By mathematical analyses, we found and proved a GHN with nonnegative weights is strictly stable. Then the GHNs with nonnegative weights is useful for associative memory. On the other hand, this kind of GHNs can be easily constructed by electronic devices. By simulation experiments on the networks with the number of neurons from 5 to 10, we found that the GHNs with simple holes are abundant and strictly stable almost. Therefore GHNs can also be effective for associative memory as Hopfield networks.

REFERENCES

Abbott, L.F. (1989). Optimal learning in neural network memories. *Journal of Physics A: Mathematics and General, 22,* L711–717.

Allen, R.B., & Alspector, J. (1990). Learning of stable states in stochastic asymmetric networks. *IEEE Transactions on Neural Networks, 1,* 233–238.

Gardner, E. (1988). The space of interactions in neural network models. *Journal of Physics A: Mathematics and General, 21,* 257–270.

Hopfield, J.J. (1982). Neural networks and physical system with emergent collective computational abilities. *Proceeding of the National Academy Sciences, USA, 79,* 2554–2558.

Ma, J. (1993). The asymmetric Hopfield model for associative memory. *Proceedings of IJCNN'93* (pp. 2611–2614). Nagoya.

Ma, J. (1995). The object perceptron learning algorithm for associative memory of asymmetric Hopfield networks. *Proceedings of ICONIP'95* (pp. 1049–1052). Beijing.

Venkatesh, S.S., & Psaltis, D. (1989). Linear and logarithmic capacities in associative memory. *IEEE Transactions on Information Theory, 35,* 558–568.

Xu, X., & Tsai, W.T. (1990). Constructing associative memories using neural networks. *Neural Networks, 3,* 301–309.