

A General Updating Rule for Discrete Hopfield-Type

Neural Network with Delay

Shenshan Qiu^{1,a}, Eric C.C. Tsang^b, Daniel S. Yeung^b, Xizhao Wang^b

^bDept. of Computing, The Hong Kong Polytechnic University

^aSouth China University of Technology, Guangzhou, 510640 China

ausqiu@scut.edu.cn; ssqiu@21cn.com, csetsang, csdaniel, csxzwang@comp.polyu.edu.hk

Abstract

In this paper, the Hopfield neural network with delay (HNND) is studied from the standpoint of regarding it as an optimized computational model. Two general updating rules for network with delay (GURD) are given based on Hopfield-type neural networks with delay for optimization problems and characterized dynamic thresholds. It is proved that in any sequence of updating rule modes, the GURD monotonously converges to a stable state of the network. The diagonal elements of the connection matrix are shown to have an important influence on the convergence process, and they represent the relationship of the local maximum value of the energy function with the stable states of the networks. All ordinary DHNN algorithms are instances of GURD. It can be shown that the convergence conditions of GURD may be relaxed in the context of applications, for instance, the condition of nonnegative diagonal elements of the connection matrix can be removed from the original convergence theorem. New updating rule mode and restrictive conditions can guarantee the network to achieve a local maximum of the energy function.

1 Introduction

Recently the convergence of various connectionist models including discrete Hopfield neural network (DHNN) [Hopfield, 1982; Wilson and Pawley, 1988; Bruck, 1990], continuous Hopfield neural network (CHNN) [Hopfield, 1982] and network with delay [Clouse and Lee, 1997] has been studied extensively. Since Hopfield and Tank [Hopfield, 1982] first applied DHNN to the traveling salesman problem, DHNN (which has a very simple and efficient hardware implementation) has demonstrated the capability to solve a wide variety of combinatorial

optimization problems. However, Hopfield and Tank's approach has been subjected to severe criticism on the quality of its solutions, especially for large-scale problems [Wilson and Pawley, 1988]. Other issues concerning DHNN such as computational complexity [Orponen, 1992], computational power, finite automata equivalency and dynamic system simulation by DHNN are beyond the scope of this paper.

It is well known that the DHNN without delay has an important property, i.e., it always converges to a stable state when operating in a serial mode and to a cycle of length at most 2 when operating in a parallel mode [Hopfield, 1982; Wilson and Pawley, 1988; Bruck, 1990; Xu *et al.*, 1998]. The problem of convergence is considered one of the most important theoretical issues in the study of the neural networks. However, in most optimization problems, the corresponding matrix and updating rule mode do not satisfy convergence conditions. Therefore its application domain is rather limited. To correct the deficiencies of the Hopfield network, various attempts such as optimal parameter selection and infeasible solution elimination have been made. Among these attempts, some are to maintain the structure of the Hopfield network, while others are to modify the network structure. In [Qiu *et al.*, 2000; Qiu *et al.*, 1999a; Qiu *et al.*, 1999b] a DHNN with delay is constructed and a serial (updating rule) mode similar to one in Hopfield network without delay is also presented. But the corresponding convergence conditions are strictly constrained. It is also shown that the energy function is dependent on the updating rule mode. It is well known that a detailed specification for the convergence of DHNN is very critical to most of the applications and it remains an open problem on how much could the convergence conditions be relaxed.

In fact, the convergence conditions, network architecture, network initial conditions as well as updating rule modes are

¹ Corresponding to: Shenshan Qiu, Department of Automatic Control Engineering, South China University of Technology, Guangzhou, 510640 China;

intimately related. The aim of this paper is to extend the convergence conditions given in [Qiu *et al.*, 2000] and to specify the updating rule mode which will make the energy function decrease monotonically with respect to the evolution time. The energy function is modified by means of a modified DHNN as well as its updating rule mode, so that the convergence of DHNN with delay is obtained.

This paper has the following organization. Section 1 is the introduction. Section 2 gives a brief review for DHNN with delay. Section 3 introduces some definitions and notations. Section 4 discusses the convergence of DHNN with delay and presents the main result. The last section concludes the paper.

2 Hopfield Networks with Delay

Discrete Hopfield neural network (DHNN) [Hopfield, 1982; Wilson and Pawley, 1988; Bruck, 1990; Xu *et al.*, 1998] is described as follows. The state of the network at time t is the vector $X(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$. In general, the state of the network at time $t+1$ is a function of $\{W, q\}$ and the state of network at time t . This network is thus completely determined by the parameters $\{W, q\}$, the initial state values $X(0)$, and the manner in which the nodes are updated (evolved). If at time step t , node x_i is chosen for updating, then at the next time step $t+1$,

$$x_i(t+1) = \text{sgn}(h_i(t)) = \begin{cases} 1 & , h_i(t) \geq 0 \\ -1 & , h_i(t) < 0 \end{cases} \quad (1)$$

where $h_i(t) = \sum_{j=1}^n w_{ij} x_j(t) - q_i$.

If only one neuron is allowed to change state at each time step by performing the state evolving equation (1) while the states of the other nodes are unchanged, then the network is said to be operating in a serial mode. If all the nodes are chosen to be updated at each time step by performing the state evolving equation (1), then the network is said to be operating in a parallel mode.

A DHNN with delay is a computational model, similar to a multilayer perceptron in which all connections are feed-forward. The difference between a DHNN with delay and a multilayer perceptron is as follows. The inputs to any node x_i for DHNN with delay may consist of the outputs of earlier nodes not only during the current step t , but also during previous d time steps $(t-1, t-2, \dots, t-d)$. Following [Bruck, 1990; Orponen, 1992], the state of a neuron is updated according to the following equation:

$$x_i(t+1) = s_i(H_i(t)) \in \{1, -1\} \quad (2)$$

where $H_i(t) = \sum_{k=0}^d \sum_{j=1}^n w_{ij}^k x_j(t-k) - q_i(t)$.

In order to simplify the analysis presented later in this paper, only the special case $d=1$, $q(t) \equiv q$ will be discussed here. $s_i(\cdot)$ represents the content of the output function of the neuron and the updating rule mode. The quadruple $N = (w^0, w^1, q, s)$ is used to represent the model (2). For instance, if $w^1 = O_{n \times n}$, $s_i(\cdot) = \text{sgn}(\cdot)$, then the quadruple $N = (w^0, w^1, q, \text{sgn})$ represents the DHNN model.

3 Updating Rule of DHNN with Delay

Let $B^n \equiv \{-1, 1\}^n$ represent the set of all vectors $X = (x_1, x_2, \dots, x_n)^T$ where each component x_i assumes the value $+1$ or -1 . We propose a new updating rule for DHNN with delay in order to extend the convergence conditions and establish an algorithm that will enable the network to escape from the local maximum points. The following is the definition of a generalized updating rule.

Definition 1. Let $N = (w^0, w^1, q, s)$ represent a DHNN with delay, starting from any initial vector $X(1) = X(0) \in B^n$. Given a sequence of neural index subsets $L(k) \subseteq \{1, 2, \dots, n\}$ for $k \geq 1$, the neural network is updated in the following form: When $k=1$, $L(1)$ is a neural index subset by means of random selection (in general $L(1) \equiv \{1, 2, \dots, n\}$), $x_i(k+1) = x_i(k)$, if $i \notin L(k)$ and for $i \in L(k)$, $k > 1$, the following defines updating rule s :

$$x_i(k+1) = \begin{cases} -x_i(k), & \text{if } x_i(k)H_i(k) > x_i(k)T_i(k) \\ x_i(k), & \text{otherwise} \end{cases} \quad (3)$$

where $H_i(k) \equiv \sum_{j=1}^n (w_{ij}^0 x_j(k) + w_{ij}^1 x_j(k-1) - q_i)$, and

$$T_i(k) \equiv -w_{ii}^0 + \sum_{\substack{j \in L(k) \\ j \neq i}} |w_{ij}^0| + 2 \sum_{j \in L(k-1)} |w_{ij}^1|$$

The updating mode is called the general updating rule for the neural network with delay, GURD for short. A sequence of neural index subsets $L(k) \subseteq \{1, 2, \dots, n\}$ satisfying the above conditions is generalized in an evolution mode. It includes a serial mode for neural network with delay when the sequence of neural index subsets is chosen which satisfy $|L(k)|=1$ [Xu *et al.*, 1998; Sun, 1998]. The following gives an illustrated example.

Example 1.1

$$w^0 = \begin{pmatrix} -7 & -2 & -2 & 4 \\ -2 & -5 & -4 & 6 \\ -2 & -4 & 11 & -35 \\ 4 & 6 & -35 & 0 \end{pmatrix}, w^1 = \begin{pmatrix} 10 & -2 & 2 & -4 \\ 1 & 5 & 2.5 & -1 \\ -2 & -2.5 & 10 & 5 \\ 4 & 1 & -5 & 10 \end{pmatrix}, q = \begin{pmatrix} -12.8294 \\ -29.4529 \\ -10.8275 \\ -4.8314 \end{pmatrix}$$

The updating results of example 1.1 are presented in Table 1. It is noted that, for each updating mode, the threshold of a neuron is different, depending on the given

sequence of neural index subsets $L(k)$ which determines the sequence of updating mode. The threshold $T_i(k)$ changes with the updating time k . However, for a Hopfield Neural network, its threshold is a constant. So the stable state of the network is constrained. In other words, when the weights and threshold parameters of a network are given, its fixed points or stable states have been fixed, and they can not change with different updating modes, such as serial mode or parallel mode. On the other hand, the stable state of GURD mode is dynamically distributed based on a given sequence of neural index subsets $L(k)$. Of course, the global maximum point will not change with respect to different updating modes. Only the local maximum points will be affected by these updating modes. Thus this dynamically distributed mechanism and delay relation enable the network to escape from local maximum points and finally reach the global maximum point (similar to LME algorithm [Peng, *et al.*, 1996], a strategy to identify which delay item can be regarded as a certain noise). This is what motivates the investigation on the updating mode and the delay neural network model.

lvs code	State vector code; Updated neuron index								SPs	E.Value
	1	2	3	4	1	2	3	4		
0→	8	12	12	13	13	13	13	13	13	Submax
1→	9	13	13	13	13	13	13	13	13	Submax
2→	10	14	14	15	15	15	15	15	15	Max
3→	11	15	15	15	15	15	15	15	15	Max
4→	4	4	4	5	5	5	5	5	5	C
5→	5	5	5	5	5	5	5	5	5	C
6→	6	6	6	6	6	6	6	6	6	B
7→	7	7	7	7	7	7	7	7	7	A
8→	8	12	12	13	13	13	13	13	13	Submax
9→	9	13	13	13	13	13	13	13	13	Submax
10→	10	14	14	15	15	15	15	15	15	Max
11→	11	15	15	15	15	15	15	15	15	Max
12→	12	12	12	13	13	13	13	13	13	Submax
13→	13	13	13	13	13	13	13	13	13*	Submax
14→	14	14	14	15	15	15	15	15	15	Max
15→	15	15	15	15	15	15	15	15	15*	Max

Table 1. Updating procedure from all initial values to SP for example 1.1

Definition 2. The function $E(u, v)$ defined below is called a bivariate energy function on B^n for $N=(w^0, w^1, q, s)$:

$$E(u, v) = u^T w^0 u + 2u^T w^1 v - 2u^T q, \quad (4)$$

where $u = x(t)$, $v = x(t-1)$. In short, the bivariate energy function is written as $E(t) \equiv E(x(t), x(t-1))$.

4 Main Results

Based on the GURD mode, the neural network with delay has the following convergence properties.

Theorem 1. Let $N=(w^0, w^1, q, s)$ be a DHNN with delay, operating in GURD mode s . Suppose w^0 is a symmetric matrix and w^1 is a diagonal dominant matrix, i.e., $w_{ii}^1 \geq \sum_{j \neq i} |w_{ji}^1|$. Then for any $k \geq 1$, $\Delta x(k) \neq 0$ or $\Delta x(k-1) \neq 0$, it is guaranteed that

$$\Delta E(k) \equiv E(x(k+1), x(k)) - E(x(k), x(k-1)) \geq 0. \quad (5)$$

For any $L(k) \subseteq \{1, 2, \dots, n\}$ the equality holds if only and if $x(k+1) = x(k) = x(k-1)$.

Proof: According to definition 2

$$E(u, v) = u^T w^0 u + 2u^T w^1 v - 2u^T q$$

where $u = x(k)$, $v = x(k-1)$, q is a threshold vector $q = (q_1, q_2, \dots, q_n)^T$. Then

$$|E(k)| \equiv |E(x(k), x(k-1))| \leq \sum_{i=1}^n \sum_{j=1}^n (|w_{ij}^0| + 2|w_{ij}^1|) + 2 \sum_{i=1}^n |q_i|.$$

For any $k \geq 2$, $i \in L(k) \subseteq \{1, 2, \dots, n\}$ and GURD in the operating mode, let $\Delta x_i(k) \equiv x_i(k+1) - x_i(k)$, $\Delta E(k) \equiv E(k+1) - E(k)$, we have

$$\begin{aligned} \Delta E(k) &\equiv x^T(k+1)w^0 x(k+1) + 2x^T(k+1)w^1 x(k) - 2x^T(k+1)q \\ &\quad - x^T(k)w^0 x(k) - 2x^T(k)w^1 x(k-1) + 2x^T(k)q \\ &= \sum_{i=1}^n 2\Delta x_i(k)H_i(x(k)) + \sum_{i=1}^n \sum_{j=1}^n (\Delta x_i(k))(\Delta x_j(k))w_{ij}^0 \\ &\quad + 2 \sum_{i=1}^n \sum_{j=1}^n x_i(k+1)\Delta x_j(k-1)w_{ij}^1 \\ &= \sum_{i=1}^n 2\Delta x_i(k) \left\{ H_i(x(k)) + \frac{1}{2} \sum_{j=1}^n (\Delta x_j(k))w_{ij}^0 + \sum_{j=1}^n (\Delta x_j(k-1))w_{ij}^1 \right\} \\ &\quad + 2 \sum_{j \in L(k-1)} \Delta x_j(k-1) \left\{ \sum_{i=1}^n x_i(k)w_{ij}^1 \right\} \\ &= \sum_{i \in L(k)} 2\Delta x_i(k) \left\{ H_i(x(k)) + \frac{1}{2} \sum_{j \in L(k)} (\Delta x_j(k))w_{ij}^0 + \sum_{j \in L(k-1)} (\Delta x_j(k-1))w_{ij}^1 \right\} \\ &\quad + 2 \sum_{i \in L(k-1)} \Delta x_i(k-1) \left\{ \sum_{j=1}^n x_j(k)w_{ji}^1 \right\} \end{aligned}$$

(case 1) If $\Delta x_i(k) > 0$, then $x_i(k) = -1$ and $x_i(k+1) = 1$.

Thus we have

$$H_i(k) > -w_{ii}^0 + \sum_{\substack{j \in L(k) \\ j \neq i}} |w_{ij}^0| + 2 \sum_{j \in L(k-1)} |w_{ij}^1|$$

being a sufficient condition for $(\Delta E(k))_i > 0$;

(case 2) If $\Delta x_i(k) < 0$, then $x_i(k) = 1$ and $x_i(k+1) = -1$.

Thus we have

$$H_i(k) < w_{ii}^0 - \sum_{\substack{j \in L(k) \\ j \neq i}} |w_{ij}^0| - 2 \sum_{j \in L(k-1)} |w_{ij}^1|$$

being a sufficient condition for $(\Delta E(k))_i > 0$;

(case 3) as w^1 is a column diagonal dominant matrix, then

$$\Delta E(k) \geq 0 \text{ when } \Delta x_i(k) = 0, \Delta x_i(k-1) \neq 0.$$

Any nonzero update with $\Delta x(k) \neq 0$ or $\Delta x(k-1) \neq 0$ in the GURD is of these three cases. So $\Delta E(k) \geq 0$.

Tables 1 and 2 present the updating results of example 1.1. Especially in Table 2, the experimental results support our main theorem. It shows that our method is more likely to converge to the maximum or the submaximum values than the Hopfield neural network method does (the Hopfield neural network could be applied to the energy function of the example 1.1, after modifying the diagonal elements of the weight matrix). The three stable points (coded SPs: 5,6,7) which are neither max nor submax value points, have the greatest energy function values.

Ivs code	Energy F.Value	Stable Points	Relation SP with (EFValue)
0	-51.8824	13	Submax
1	-58.5569	13	Submax
2	29.4276	15	Max
3	-5.24686	15	Max
4	69.9292	5	C
5	111.255	5	C
6	119.239	6	B
7	132.565	7	A
8	3.43518	13	Submax
9	28.7607	13	Submax
10	68.7452	15	Max
11	66.0707	15	Max
12	101.247	13	Submax
13	174.572	13	Submax
14	134.557	15	Max
15	179.882	15	Max

Table 2. Updating results to all initial values for example 1.1

The following example 1.2 is a revised version of example 1.1. After modifying the original threshold vector (by adding a certain noise or standard Gaussian noise) and the diagonal elements of the weight w^0 . The updating results of example 1.2 are given in Table 3. The experimental results of example 1.2 show a much better performance than example 1.1. There are only two stable points. SP 13 is the maximum value point and SP 15 is the submaximum value point.

Example 1.2

$$w^0 = \begin{pmatrix} 0 & -2 & -2 & 4 \\ -2 & 0 & -4 & 6 \\ -2 & -4 & 0 & -3.5 \\ 4 & 6 & -3.5 & 0 \end{pmatrix}, w^j = \begin{pmatrix} 10 & -2 & 2 & -4 \\ 1 & 5 & 2.5 & -1 \\ -2 & -2.5 & 10 & 5 \\ 4 & 1 & -5 & 10 \end{pmatrix}, q = \begin{pmatrix} -33.9882 \\ -40.3587 \\ -8.9825 \\ -4.99416 \end{pmatrix}$$

It is no surprise that the convergence performance of example 1.2 is better than that of example 1.1. This shows that by adjusting the original threshold values of the network parameters, one can change or decrease the local maximum value points. However, this improvement comes as a result

of 20 times updating tests and 50 random selection of parameter threshold values. It remains an important open problem on how such threshold values can be adjusted to produce a better performance.

Ivs code	Energy F. Value	Stable Points	E.Value
0	-111.647	13	Max
1	-117.671	13	Max
2	-37.7172	15	Submax
3	-71.7406	15	Submax
4	53.7878	13	Max
5	95.7644	13	Max
6	95.7178	15	Submax
7	109.694	15	Submax
8	28.3057	13	Max
9	54.2822	13	Max
10	86.2356	15	Submax
11	84.2122	15	Submax
12	169.741	13	Max
13	243.717	13*	Max
14	195.671	15	Submax
15	241.647	15*	Submax

Table 3. Updating results to all initial values for example 1.2

All the updating rules are summarized in Table 4. The following illustration is used to explain the entries in Table 4.

Ivs code	State vector code/updated neuron index								SPs	E.Value
	1	2	3	4	1	2	3	4		
0→	8	12	12	13	13	13	13	13	13	Max
1→	9	13	13	13	13	13	13	13	13	Max
2→	10	14	14	15	15	15	15	15	15	Submax
3→	11	15	15	15	15	15	15	15	15	Submax
4→	12	12	12	13	13	13	13	13	13	Max
5→	13	13	13	13	13	13	13	13	13	Max
6→	14	14	14	15	15	15	15	15	15	Submax
7→	15	15	15	15	15	15	15	15	15	Submax
8→	8	12	12	13	13	13	13	13	13	Max
9→	9	13	13	13	13	13	13	13	13	Max
10→	10	14	14	15	15	15	15	15	15	Submax
11→	11	15	15	15	15	15	15	15	15	Submax
12→	12	12	12	13	13	13	13	13	13	Max
13→	13	13	13	13	13	13	13	13	13*	Max
14→	14	14	14	15	15	15	15	15	15	Submax
15→	15	15	15	15	15	15	15	15	15*	Submax

Table 4. Updating procedure from all initial values to SPs for example 1.2

Let the initial value $X(1) = X(0)$ be $(-1, -1, -1, -1)^T \in B^4$. The evolving operational process can be interpreted as follows. First choose a sequence of neural index subsets $L(1) = \{1\}$,

$$L(2) = \{2\}, \quad L(3) = \{3\}, \quad L(4) = \{4\}, \quad L(k) = L(i), \\ i = k - [(k-1)/4] \cdot 4, \quad k > 4,$$

where $[s]$ denotes the largest integer which is smaller than s ; bivariate $\{st_code, EFvalue = m_i\}$ represents the neuron state vector code(st_code) and the corresponding energy function value($EFvalue$); and $\xrightarrow{L(i)}$ means updating all elements of the neural index subsets in $L(i)$.

Initial value $\{0, EFvalue = -111.647\} \xrightarrow{L(1), L(2), L(3), L(4)} \{13, EFvalue = 243.717\} \xrightarrow{L(4)} \dots$ It converges to the stable state 13.

Theorem 2. Let $N = (w^0, w^1, q, s)$ be a DHNN with delay, operating in a GURD mode. Suppose w^0 is a symmetric matrix and w^1 is a diagonal dominant matrix, i.e., for all i , $w_{ii}^1 \geq \sum_{j \neq i} |w_{ji}^1|$. Then for any $k \geq 1$, we have

$$\Delta E(k) \equiv E(u, u) - E(u, v) \geq 0, \text{ for } u, v \in B^n, \quad v \neq u \quad (6)$$

The inequality holds if only and if $\bigcap_{k=1}^{\infty} L(k)$ is a nonempty set, and there is an index $i \in \bigcap_{k=1}^{\infty} L(k)$ such that $w_{ii}^1 > \sum_{j \neq i} |w_{ji}^1|$, where $u = x(k+1)$, $v = x(k)$.

Proof: According to definition 2

$$E(u, v) = u^T w^0 u + 2u^T w^1 v - 2u^T q,$$

where $u = x(k)$, $v = x(k-1)$, and q is a threshold vector $q = (q_1, q_2, \dots, q_n)^T$. Let $\Delta x_i(k) \equiv x_i(k+1) - x_i(k)$, $\Delta E(k) \equiv E(k+1) - E(k)$. Then

$$\begin{aligned} \Delta E(k) &\equiv 2x^T(k+1)w^1x(k) - 2x^T(k+1)q \\ &\quad - 2x^T(k)w^1x(k-1) + 2x^T(k)q \\ &= 2x^T(k)w^1\Delta x(k-1) + 2\sum_{i=1}^n \Delta x_i(k-1) \left\{ \sum_{j=1}^n w_{ji}^1 x_j(k) \right\}. \end{aligned}$$

Without loss of generality, we assume that $L(k-1) \equiv \{1, 2, \dots, n\}$, $\Delta x_i(k-1) \neq 0$, $i \in L(k-1)$. Since w^1 is a column diagonal dominant matrix, it implies that $\Delta E \geq 0$.

If $\bigcap_{k=1}^{\infty} L(k)$ is a nonempty set of $\{1, 2, \dots, n\}$, w^1 is a

diagonal dominant matrix and there is an index $i \in \bigcap_{k=1}^{\infty} L(k)$

such that $w_{ii}^1 > \sum_{j \neq i} |w_{ji}^1|$, we can conclude

$$\Delta E \equiv E(u, u) - E(u, v) > 0.$$

On other hand, if $\Delta E \equiv E(u, u) - E(u, v) \leq 0$, i.e.,

$$\begin{aligned} 0 \geq \Delta E &= u^T w^1(u-v) = 2 \left\{ 2 \sum_{i=1}^n w_{ii}^1 + \sum_{j=1}^n \sum_{i \neq j} u_i w_{ij}^1 (u_j - v_j) \right\} \\ &\geq 4 \sum_{i=1}^n (w_{ii}^1 - \sum_{j \neq i} |w_{ji}^1|) \geq 0 \end{aligned}$$

Since w^1 is a diagonal dominant matrix, hence $w_{ii}^1 = \sum_{j \neq i} |w_{ji}^1|$ is satisfied for all $i \in \{1, 2, \dots, n\}$. It is a contradiction as there is an index i such that $w_{ii}^1 > \sum_{j \neq i} |w_{ji}^1|$. This completes the

proof of theorem 2.

From the general updating rule (5) in theorem 1, it is noted that the network trajectory evolves in such a way that the energy function is always nondecreasing; that is,

$$\Delta E(k) \equiv E(x(k+1), x(k)) - E(x(k), x(k-1)) \geq 0$$

Using the inequality (6) in theorem 2

$$E(x(k+1), x(k+1)) - E(x(k+1), x(k)) \geq 0.$$

Hence

$$E(x(k+1), x(k+1)) \geq E(x(k+1), x(k)) \geq E(x(k), x(k-1)) \quad (7)$$

The network trajectory now has a new step, i.e., at the $(k+1)th$ step, instead of updating from the kth step with initial values $x(k), x(k-1)$, now updating from the kth step with initial values $x(k), x(k)$. The modification allows the energy function to accelerate the rate of convergence and simplify the GURD. In the following, we define a new GURD rule.

Definition 3. Let $N = (w^0, w^1, q, s)$ represent DHNN with delay, starting from any initial vector $X(1) = X(0) \in B^n$, given a sequence of neural index subsets $L(k) \subseteq \{1, 2, \dots, n\}$ for $k \geq 1$, the neural network is updated as follows:

- 1) When $k = 1$, $L(1) \subseteq \{1, 2, \dots, n\}$ is a neural index subset by means of random selection (in general $L(1) \equiv \{1, 2, \dots, n\}$),
- 2) For $L(k) \subseteq \{1, 2, \dots, n\}$, if $i \notin L(k)$, let $x_i(k+1) = x_i(k)$ and for $i \in L(k)$. The updating rule s is given by:

$$x_i(k+1) = \begin{cases} -x_i(k), & \text{if } x_i(k) \left(w_{ii}^0 + 2 \sum_{\substack{j \in L(k) \\ j \neq i}} |w_{ij}^0| \right) > x_i(k) H_i(k) \\ x_i(k), & \text{otherwise} \end{cases} \quad (8)$$

where $H_i(k) = \sum_{j=1}^n (w_{ij}^0 x_j(k) + w_{ij}^1 x_j(k-1) - q_i)$.

This updating mode is also called the simplified general updating rule for neural network with delay, SGURD for short.

It is noted that both w^0 being a symmetric matrix and w^1 being a diagonally dominant matrix are required for the

SGURD. Comparing the two constrained conditions (6) and (8), one may easily conclude that the range region of x invariable condition (otherwise case in (8)) decreases. It means that the energy function can reach the local maximum value in a short updating time. In other words, we only need to execute the SGURD rule $2n+1$ steps at most [Qiu *et al.*, 1999a]. But for the Hopfield network, the updating rule may require up to n^2 steps in order to obtain a stable state when the energy function has reached to a local maximum [Bruck, 1990].

It is noted that theorem 2 generalizes the results in [Qiu *et al.*, 2000; Qiu *et al.*, 1999a; Qiu *et al.*, 1999b] which extend the constrained conditions of the weight matrices w^0 , w^1 . The requirement that the diagonal elements in w^0 are non-negative (serial mode) and w^0 is nonnegative definite (parallel mode) are not necessary for the GURD mode. It has been shown that the updating rule process of the network and the activation process of the neurons are characterized by a dynamical process. But the sigmoid function still remains as the most popular choice for the network's activation function, and the serial, parallel and partially parallel modes are considered as important updating rule modes. There are no other modes.

Corollary 2.1 After a finite number of updates, under the GURD mode, the neural network with delay will converge to a fixed point.

In fact the fixed point is a local minimum point of the energy function. On the other hand, the convergence property is indispensable for any practical application design of neural network with delay.

5 Conclusion

In this article we have demonstrated how a generalized updating rule can cause the energy function to increase monotonously for neural network with delay. The GURD mode can operate in any sequence of updating modes. We have shown that the neural network with delay converges when operating in the GURD mode. The algorithm presented in [Qiu *et al.*, 2000; Qiu *et al.*, 1999a; Qiu *et al.*, 1999b], including the original MHNN algorithm and the partially parallel mode algorithm [Hopfield, 1982; Wilson and Pawley, 1988; Bruck, 1990; Xu *et al.*, 1998; Sun, 1998; Lee, 1999], can be easily derived from our main result which is supported by the experimental results of examples 1.1 and 1.2.

Acknowledgements

The authors would like to thank reviewers for their comments and suggestions which helped in improving the manuscript. This research was partially supported by the national nature sciences foundation, China (Grant:69934030) and nature sciences foundation of South China of University Technology.

References

- [Hopfield, 1982] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA*, 1979: 2554-2558, 1982.
- [Wilson and Pawley, 1988] G. N. Wilson, and G. S. Pawley. On the Stability of the Traveling Salesman Problem Algorithm of Hopfield and Tank. *Biological Cybernetics*, 58:63-70, 1988.
- [Bruck, 1990] Bruck J, On the convergence properties of the Hopfield model. *Proceedings of IEEE*, 78(10):1579-1585, 1990.
- [Clouse and Lee, 1997] Daniel S. Clouse and Giles C. Lee. Time-delay neural network: Representation and induction of finite-state machines. *IEEE Transactions on Neural Networks*, 8(5):1065-1070, 1997.
- [Orponen, 1992] P. Orponen. Neural networks and complexity theory. *Proc. 17th International Symposium on Mathematical Foundations of Computer Science*, pages 50-61, Lecture Notes in Computer Science 629, Springer-Verlag, Berlin, Germany, 1992.
- [Xu *et al.*, 1998] Zongbin Xu, Guoqing Hu and Chungping Kwong. Asymmetric Hopfield-type networks: theory and applications. *Neural Networks*, 9(3):483-501, 1996.
- [Sun, 1998] Yi Sun. A generalized updating rule for modified Hopfield neural network for quadratic optimization. *Neurocomputing*, 19:133-143, 1998.
- [Qiu *et al.*, 2000] Shenshan Qiu, Eric C.C. Tsang and Daniel S. Yeung. Stability of discrete Hopfield neural networks with time-delay, In *Proceedings of the 2000 IEEE International Conference on system, Man Cybernetics*, pages 2545-2550, Nashville, Tennessee, October, 2000.
- [Qiu *et al.*, 1999a] Shenshan Qiu, Xiaofei Xu, Mingzhu Liu and Yadong Wang. Convergence of discrete Hopfield-type neural network with time-delay in a serial mode. *Journal of Computer Research & Development* 36 (5):557-563, 1999.
- [Qiu *et al.*, 1999b] Shenshan Qiu, Xiaofei Xu, Chunsheng Li and Mingzhu Liu. Matrix criterion of Dynamic analysis in discrete neural networks with delay. *Journal of Software*, 10(10):1108-1113, 1999.
- [Lee, 1999] Donq-liang Lee. New stability conditions for Hopfield networks in partial simultaneous update mode. *IEEE Transactions on Neural Networks*, 10(4):975-978, 1999.
- [Peng, *et al.*, 1996] Mengkang Peng, Narendra K. and Allistair F. Armitage. An investigation into the improvement of local minima of the Hopfield network. *Neural Networks*, 9(7):1241-1253, 1996.