Improved Hopfield Networks for Pattern Sequence Recognition

Donq-Liang Lee

Department of Electronics Engineering,
Ta-Hwa Institute of Technology,
Chung-Lin, Hsin-Chu, Taiwan 307, R.O.C.
E-mail:etldl@et4.thit.edu.tw

Abstract

This paper presents a novel continuous-time Hopfield-type network which is suitable for temporal sequence recognition. Since it is difficult to implement a desired flow vector field distribution by using conventional matrix encoding scheme, a time-varying Hopfield model (TVHM) is proposed. The weight matrix of the TVHM is constructed in such a way that its auto-correlation and cross-correlation parts are encoded from two different sets of patterns. The proposed approach is different from the existing methods because neither synchronous dynamics nor interpolated training patterns are required. Experimental results are presented to illustrate the validity, recall capability, and the applications of the proposed model.

Keywords: Hopfield networks, recalling dynamics, auto-correlation, cross-correlation, pattern sequence recognition

1 Introduction

Associative neural networks, characterized by information storage and recall, have given rise to much interest in recent years, because of their wide range of applications in areas such as content addressable memory (CAM) and pattern recognition [1],[2],[12],[13]. Pattern recognition can be divided into two cases, i.e., static pattern recognition and pattern sequence recognition. In the first case each training pattern is stored as a stable (static) attractor in a corresponding energy landscape. In the second case each sequence of patterns is stored as trajectory attractors in the energy landscape. In the past fifteen years several design techniques have been proposed for static pattern recognition [3]-[5],[12],[13]. However, the possibility of storing/recalling pattern sequences by neural networks has received little attention in literature.

In the recent work of Morita [8], the fundamental problems of recalling pattern sequences by neural networks are investigated. As pointed out by Morita, when pattern sequences are to be stored, conventional neural networks do not work well unless discrete *synchronous* dynamics is used (for a detail of storing/recalling pattern sequences in discrete synchronous networks, please refer to [6],[11],[13],[14]). Since flow vectors [8] between two adjacent stored attractors do not coincide in direction, asynchronous network dynamic is not suitable for pattern sequence recall .Sompolinsky and Kanter [9] proposed a continuous-time asymmetric neural model (see also [10]). In their model two kinds of weight matrix are used, one for auto-association and the other for hetero-association encoding.. The latter is referred to as a synchronizing mechanism to achieve recall between stored patterns. However, it may not work well if the network starts with an initial state which locates midway between two adjacent stored patterns. Morita [8] proposed the nonmonotone neural networks with improved continuous dynamics. With his model, a pattern sequence can be recalled in such a way that network state changes gradually from one stored pattern to another. However, his method requires a proper overlap between adjacent stored patterns. Otherwise, some interpolated patterns must be added between these patterns to ensure successful in recall.

A new weight matrix formulation ensuring the storage and recall of pattern sequences by a Hopfield network is investigated in this paper. Since it is difficult to implement a desired flow vector field distribution by using conventional matrix encoding scheme, a time-varying Hopfield model (TVHM) is proposed. With this concept, a network can recall pattern sequences in such a way that neither synchronous dynamics nor interpolated training patterns are required. The validity, capacity, recall capability, and the applications of the proposed model are verified by extensive experimental results.

2 Conventional Temporal Association Geometry

Consider a Hopfield model [2] consisting of N two-state neurons $\{z_i\}_{i=1}^{N}$, where $z_i \in \{-1,+1\}$ represents the state of the ith neuron. Let T denotes the matrix of the synaptic weights of this model. Then the conventional evolution (update) equation of a discrete *synchronous* Hopfield model can be represented as

$$Z(t+1) = sgn\{TZ(t)\} \qquad (1)$$

where $Z(t) = (z_1(t), z_2(t), ..., z_N(t))^T$ is the state (at time t) of the network, The word "*synchronous*"means all the neurons are updated simultaneously or in parallel.

Suppose that a sequence of patterns

$$U^1 \to U^2 \to \ldots \to U^p \tag{2}$$

is to be stored in the network. These patterns have been properly numbered according to their orders in the sequence. They are stored in the network by using the *cross-correlation* weight matrix [8], i.e.,

$$T = \frac{1}{N}\sum_{\mu=1}^{p-1} U^{\mu+1}(U^{\mu})^T \tag{3}$$

Here the p patterns $U^1, U^2, \ldots, U^p \in \{-1,+1\}^N$ are assumed to be orthogonal (or near orthogonal) to each other. As pointed out by Morita [8], if $Z(0)$ is close to U^1, it is desired that $Z(1) = U^2$, $Z(2) = U^3$, ..., i.e., the stored sequence is recalled. However, recall is always impossible if one starts with a $Z(0)$ which located at the midway between U^μ and $U^{\mu+1}$. In this case the flow at $Z(0)$ points not to $U^{\mu+1}$but between $U^{\mu+1}$and $U^{\mu+2}$. This is the reason that the cross-correlation matrix memory is only suitable for discrete *synchronous* dynamics (see Fig. 1 for a schematic representation of the flow distribution). If *asynchronous (* i.e., only one neuron is allowed to change state at any time instant) or continuous dynamics are desired, the state changes gradually from $Z(0)$ to $U^{\mu+1}$. To prevent the case that Z gradually varying its direction of movement, it is necessary that the flow , alone the path between U^μ and $U^{\mu+1}$, point to $U^{\mu+1}$. That is, the desired flow vector field in the state space should be distributed as in Fig. 2. However, it is difficult to implement such flow vector field distribution by using the static matrix in (3).

3 A Hopfield Model with a Time -Varying Weight Matrix

Consider an N-neuron continuous-time network whose dynamics are described by

$$\tau\frac{dz_i}{dt} = -z_i + \sum_{j=1}^{N} t_{ij}sgn(z_j), \qquad i = 1, 2, \ldots, N \tag{4}$$

where z_i is the input to neuron i, t_{ij} is the connection weight between the ith and the jth neurons. Furthermore, $\tau > 0$ is a time constant. Here the weight matrix $T = [t_{ij}]$ is considered as a time-varying function of the form

$$T(t) = \lambda(t)M_1 + (1-\lambda(t))M_2, \tag{5}$$

in which

$$M_1 = \frac{1}{N}\left(\sum_{\mu\in ODD} U^\mu(U^\mu)^T + \sum_{\mu\in EVEN} U^{\mu+1}(U^\mu)^T\right), \tag{6a}$$

and

$$M_2 = \frac{1}{N}\left(\sum_{\mu\in EVEN} U^\mu(U^\mu)^T + \sum_{\mu\in ODD} U^{\mu+1}(U^\mu)^T\right). \tag{6b}$$

In (6) ODD and $EVEN$ denote the sets of pattern indices corresponding to odd and even numbers, respectively. A pattern is said to be an *even (odd)* pattern if its pattern index $\mu \in EVEN$ (ODD). The matrix $T(t)$ is made up of two static (constant) matrices: $M_1$and M_2. In (5) $\lambda(t)$ is a time-varying scaling factor function, $0 \le \lambda(t) \le 1$. The weight matrix $T(t)$ is organized by (5) so that its components are determined from the weighted average of $M_1$and M_2. As seen from (5), if $\lambda(t)$ approaches 1 and 0, $T(t)$ will be dominated by M_1 and M_2, respectively. The model is called the time-varying Hopfield model (TVHM) here for its time-varying weight matrix.

$M_1$and M_2 in (6) are constructed in such a way that their auto-correlation and cross-correlation parts are encoded from two different sets of patterns. For example, the cross-correlation part of $M_1$is encoded by the even patterns whereas the auto-correlation part is encoded by the odd patterns. The matrix M_2 is constructed in a similar but contrary way. In fact, M_1 and M_2 are complements for each other. First consider an extreme case where $\lambda(t) \to 1$. The matrix $T(t)$ is dominated by M_1. In this case the flow at each even pattern U^μ will point to the next pattern $U^{\mu+1}$. In contrary the flow at each odd pattern will point to itself. In the other case where $\lambda(t) \to 0$, a similar but contrary result is obtained. The flow vector field distribution of the two cases is schematically represented by Fig. 3 and Fig. 4. The idea behind the formulation of $T(t)$ is that flow vectors between any two adjacent stored patterns (if they exist) are of the same directions. Furthermore, with the aid of the time-varying scaling factor $\lambda(t)$, the flow vector field distribution around a stored pattern can be modulated by the time variable t.

The following two theorems emphasis how $\lambda(t)$ modulates the directions of the flow vectors within the state space.

Theorem 1: Suppose that the Hamming distance between a state Z and a stored pattern U^{μ^*} is $H(Z, U^{\mu^*})$ and

$$H(Z, U^{\mu^*}) \le \frac{N}{2p}\left(|2\lambda - 1| - \sum_{\mu\ne\mu^*}^{p} |\alpha_{\mu^*\mu}|\right) \equiv r^{\mu^*}(\lambda) \tag{7}$$

where $\alpha_{\mu^*\mu}$ denotes the normalized correlation or overlap between $U^{\mu^*}$and a different stored pattern U^μ, that is,

$$\alpha_{\mu^*\mu} = (1/N)(U^{\mu^*})^T U^\mu. \tag{8}$$

According to the state equation (4), the flow vector at Z will point to

$$\text{(a) } U^{\mu^*} \text{ if } \begin{cases} \lambda \to 1, \text{ and } \mu^* \in ODD \\ \lambda \to 0, \text{ and } \mu^* \in EVEN \end{cases}$$

or

(b) U^{μ^*+1} if $\begin{cases} \lambda \to 1, \text{ and } \mu^* \in EVEN \\ \lambda \to 0, \text{ and } \mu^* \in ODD \end{cases}$

Proof: see Appendix I.
From theorem 1, the following facts can be deduced.

i) Flow vector field in Fig. 3 and Fig. 4 can be realized if $\lambda \to 1$ and $\lambda \to 0$, respectively.

ii) $r^{\mu^*}(\lambda)$ can be referred to as the *radii of attraction* (around the pattern U^{μ^*}) and is determined by the correlations $\{\alpha_{\mu^*\mu}\}_{\mu=1,\mu\neq\mu^*}^{p}$ and the scaling factor $\lambda(t)$. In other words, the size of a valid $r^{\mu^*}(\lambda)$ can be modulated by $\lambda(t)$ as long as $\sum_{\mu\neq\mu^*}^{p} |\alpha_{\mu^*\mu}| < 1$.

It is desired that the network starts with an initial state and visits the transient (unstable) attractors one by one till the final (stable) attractor is reached. Flow vector field distribution must be altered back and forth between those depicting in Fig. 3 and Fig. 4. For this reason, $\lambda(t)$ must be assigned as a periodical function of time and its value will change between 0 and 1. Though there are many candidates for $\lambda(t)$, without loss of generality, only the triangular wave function is discussed in the rest of this paper. Its waveform is depicted in Fig. 5.
The height h and the period of the waveform P have close relation with the recalling performance of the network. To emphasize this, the following theorem is proposed.
Theorem 2: Once the network reaches a stored pattern U^{μ^*}, its state will not change as long as $\lambda > \eta(\mu^*)$ (if $\mu^* \in ODD$) or $\lambda < \eta(\mu^*)$ (if $\mu^* \in EVEN$) where

$$\eta(\mu^*) = \begin{cases} \max_k\{\eta_k(\mu^*)\} & \text{if } \mu^* \in ODD \\ \min_k\{\eta_k(\mu^*)\} & \text{if } \mu^* \in EVEN \end{cases} \qquad (9)$$

and

$$\eta_k(\mu^*) = \frac{\sum_{\mu\in ODD} u_k^{\mu+1}\alpha_{\mu^*\mu} + \sum_{\mu\in EVEN} u_k^{\mu}\alpha_{\mu^*\mu}}{\sum_{\mu=1}^{p}(-1)^{\mu}(u_k^{\mu} - u_k^{\mu+1})\alpha_{\mu^*\mu}} \qquad (10)$$

Proof: see Appendix II.
The description in theorem 2 implies that the time duration of the transient (unstable) attractors can be controlled once the variation of $\lambda(t)$ is under controlled. In other words, the time the system stayed in a specific state U^{μ^*} is determined proportionally by the time of which the magnitude of λ satisfies $0 \leq \lambda < \eta(\mu^*)$ or $\eta(\mu^*) < \lambda \leq 1$. Moreover, since each stored pattern has to be visited within a nonzero time interval, the height of the triangular wave must satisfy:

$$h > 2\max\{\lambda_{\max} - 0.5,\ 0.5 - \lambda_{\min}\}, \qquad (11)$$

where

$$\begin{cases} \lambda_{\max} = \max_{\mu^*\in ODD}\{\eta(\mu^*)\} \\ \lambda_{\min} = \min_{\mu^*\in EVEN}\{\eta(\mu^*)\} \end{cases} \qquad (12)$$

This property is schematically represented in Fig. 5.
Fig. 6-Fig. 8 show examples of the TVHM recalling procedure. The stored pattern sequence is generated by random (each u_k^{μ} takes the values +1 or −1 with equal probabilities), and the experimental conditions are: $p = 6$, $N = 100$, and $\tau = 10$. The objective is to store the following relation (a transient sequence of states leading to an attractor):

$$U^1 \to U^2 \to \ldots \to U^p \longleftrightarrow U^p$$

by a TVHM. It is desired that U^p is the only stable attractor, while the others are all transient (unstable) attractors. Let $U^{p+1} = U^p$ be an augmented vector, the sequence can be stored in M_1 and M_2 by (6) for $\mu = 1,\ldots, p, p+1$. This is because U^p will be the stable attractor if $U^{p+1} = U^p$. In order to simulate the dynamics by a computer, here a discrete approximation of (4) is applied. Substituting $dz_i/dt = \{z_i(t+\Delta t) - z_i(t)\}/\Delta t$ into (4), the following equation is obtained:

$$z_i(t+\Delta t) = (1-\Delta t/\tau)z_i(t) + (\Delta t/\tau)\sum_{j=1}^{N} t_{ij}\, sgn(z_j), \qquad (13)$$

Fig. 6 ~Fig. 9 are obtained by setting $\Delta t = 1$, the period of triangular wave P is set to 6τ in Fig. 6. The magnitude of P should be chosen carefully. In general, if P small the system dynamics (13) may not catch up with the variation of t (see Fig. 7). On the other hand, a too large P may reduce the time duration of the transient (see Fig. 8). By experiment, the typical values are found to be $4\tau \leq P \leq 15\tau$.
Fig. 6 and Fig. 9 demonstrate how h modulates the system dynamics. In this example, $\lambda_{\max} = 0.5652$ and $\lambda_{\min} = 0.3059$. A perfect recall can be expected if $\lambda_{\max} < 0.5 + h/2$ and $\lambda_{\min} > 0.5 - h/2$ (ref. Fig. 5 & (11)). Fig. 9 shows that it is possible to produce an imperfect recall if one use a triangular wave with height $h = 0.3 > 2(0.5 - \lambda_{\min}) = 0.2882$.

4 A Way of Increasing the Storage Capacity of the TVHMs

M_1 and M_2 obtained by (6) may fail to produce correct sequence when p is large. Due to the use of outer-product type of correlation matrices, the storage capacity of (6) is very low. However, since the problem can be transformed to a linear separation problem (or a perceptron problem [15]). There are many alternative methods, such as the *perceptron algorithm* (Minsky and Papert, [15]), the *pocket algorithm* (Gallent, [16]), the *H-K rule* (Ho and Kashyap, [17]), the *minimum-overlap algorithm* (Krauth and Mezard, [18]), and the *pseudo-inverse technique* [19], which can be utilized to improve the capacity in certain degree.The following illustration shows how one can learn M_1 and M_2 by means of the *pseudo-inverse* *technique(PIT,* [19]*)*. Note that the problem can be rewritten as finding M_1 and M_2 such that

$$M_1U^\mu = \begin{cases} U^\mu & \text{if } \mu \in ODD \\ U^{\mu+1} & \text{if } \mu \in EVEN \end{cases}, \quad \mu = 1,\ldots,p; \tag{14}$$

and

$$M_2U^\mu = \begin{cases} U^\mu & \text{if } \mu \in EVEN \\ U^{\mu+1} & \text{if } \mu \in ODD \end{cases}, \quad \mu = 1,\ldots,p. \tag{15}$$

Without loss of generality, let $U^{p+1} = U^p$ such that the final pattern U^p will be a stable attractor. Then (14) and (15) can be further reduced to a pair of matrix equations

$$M_1\Sigma = A; \tag{16a}$$

and

$$M_2\Sigma = B, \tag{16b}$$

where

$$\Sigma = (U^1, U^2, \ldots, U^p), \tag{17}$$

$$A = \begin{cases} (U^1, U^3, U^3, \ldots, U^{p-1}, U^{p-1}, U^p) & \text{if } p \text{ even} \\ (U^1, U^3, U^3, \ldots, U^p, U^p) & \text{if } p \text{ odd} \end{cases} \tag{18}$$

and $B =$

$$\begin{cases} (U^2, U^2, U^4, U^4, \ldots, U^p, U^p) & \text{if } p \text{ even} \\ (U^2, U^2, U^4, U^4, \ldots, U^{p-1}, U^{p-1}, U^p) & \text{if } p \text{ odd} \end{cases} \tag{19}$$

Note that $A, B, \Sigma \in \{-1, 1\}^{N\times p}$. The solution of (16) will be

$$\begin{cases} M_1 = A\Sigma^+, \\ M_2 = B\Sigma^+, \end{cases} \tag{20}$$

where Σ^+ is the pseudo-inverse [14] of Σ . For example, if Σ has full column rank, i.e., $(\Sigma\Sigma^T)$ is invertible, Σ^+ can be computed by using a simple matrix equation, e.g., $\Sigma^+ = \Sigma^T(\Sigma\Sigma^T)^{-1}$. When the columns of Σ are linearly dependent, Σ^+ can be obtained by using the singular value decomposition (SVD) approach. In the latter case $M_1 = A\Sigma^+$ is an approximate solution of (16a) which minimizes the error $E = \|A - M_1\Sigma\|$ ($\|\cdot\|$ denotes the Euclidean norm operator).

5 A Pattern Sequence Recognition Example

This example shows that the TVHM is capable of producing *many* temporal evolutions by only two weight matrices M_1 & M_2. The objective is to train a TVHM to perform the three sequences shown in Fig. 10. Here all patterns are composed of 9×9 small pixels corresponding to vectors in $\{-1, 1\}^{81}$. The first row in Fig. 10 represents digits "1" to "4"; the second & third rows are the corresponding numbers in Chinese and Roman representations, respectively. The weight matrices M_1 and M_2 are obtained by using (20). After learning has been performed, the initial state of the TVHM was applied externally, and the sampled state variation $Z(0)$, $Z(4)$, $Z(8)$, ..., $Z(92)$, was recorded. In this example, the experimental conditions are: $P = 4\tau$, $h = 0.9$, $\tau = 10$, and $\Delta\, t = 1$. Table I shows the results of successful pattern sequence recalling when noisy patterns, $\tilde{U}^1$, $\tilde{U}^5$, and $\tilde{U}^9$ are used as initial states of the TVHM. The noise was created by randomly inverting each bit from $+1$ to -1 or vice versa with a specific Hamming distance. For a noisy input, if the corresponding stored stable attractor (pattern) is recalled, it is a success; otherwise it is a failure. The percentage of successful recall is summarized in Table I where each datum is obtained from average of 100 independent trial runs. With 20, 12, and 12 noisy pixels, the TVHM attains approximately 75% correct recalls for sequences, $\tilde{U}^1 \to U^4$, $\tilde{U}^5 \to U^8$, and $\tilde{U}^9 \to U^{12}$, respectively. One example of the recall procedure is depicted in Fig. 11.

6 Conclusions

In this paper, a possible way of realizing pattern sequence recognition using TVHM is proposed. The training pattern sequence is stored in a TVHM by using two different but complementary encoded weight matrices. The model is operated with continuous-time dynamics in which the weight matrix is varied with time. Theorems regarding the *radii of attraction* and the recalling dynamics of the HTVHM have been investigated. On realizing sequence recognition, the proposed TVHM possesses advantages over the existing neural models: (i) the structure is simpler than those models which incorporate delay mechanisms.(ii) the training set is original; no additional interpolated patterns required. The validity, recall capability, and the applications of the proposed TVHM are verified by computer simulations.

Acknowledgement

This work was supported by the National Science Councile, Taiwan, Republic of China under the Grant NSC88-2213-E-233-001

Appendix I

Proof of theorem 1: Let Z be a state in the defined state space S. The distribution of flow vectors in S is tschematically represented by Fig. 3 and Fig. 4 . Flow at any arbitrary state Z is nearly parallel with the vector $V(Z) \equiv T(t)Z$. By (5) and (6), this vector will be given by $V(Z) = T(t)Z =$

$$\frac{1}{N}\Bigg(\lambda \sum_{\mu\in ODD} U^\mu\left[(U^\mu)^T Z\right] \tag{22}$$
$$+\lambda \sum_{\mu\in EVEN} U^{\mu+1}\left[(U^\mu)^T Z\right]$$
$$+(1-\lambda) \sum_{\mu\in EVEN} U^\mu\left[(U^\mu)^T Z\right]$$
$$+(1-\lambda) \sum_{\mu\in ODD} U^{\mu+1}\left[(U^\mu)^T Z\right]\Bigg)$$

Now consider the following different cases:

A. If $\lambda \to 1$

Case 1: Assume Z is closed to a stored pattern U^{μ^*} where μ^* is an odd number. Then flow at Z will point to U^{μ^*} if

$$[T(t)Z]_k u_k^{\mu^*} \geq 0, \forall\, k$$

, i.e., vector $sgn[T(t)Z]$ and $U^{\mu^*}$are of the same directions. Let D_μ be the Hamming distance between Z and a stored pattern U^μ, $D_\mu = H(\ Z\ ,\ U^\mu)$, from (22),

$$\begin{aligned}
&[T(t)Z]_k u_k^{\mu^*} \\
\geq\ & \frac{\lambda}{N}\left((N-2D_{\mu^*}) - \sum_{\mu\in ODD,\mu\neq\mu^*} |N-2D_\mu|\right. \\
&\left. - \sum_{\mu\in EVEN} |N-2D_\mu|\right) + \frac{(1-\lambda)}{N} \\
&\left(- \sum_{\mu\in EVEN} |N-2D_\mu| - \sum_{\mu\in ODD} |N-2D_\mu|\right) \\
=\ & \frac{\lambda}{N}\left((N-2D_{\mu^*}) - \sum_{\mu\neq\mu^*}^{p} |N-2D_\mu|\right) \quad (23) \\
& -\frac{(1-\lambda)}{N}\sum_{\mu=1}^{p}|N-2D_\mu| \\
\equiv\ & a
\end{aligned}$$

Let $\alpha_\mu = (1/N)(U^\mu)^T Z$ and $\alpha_{\mu^*\mu} = (1/N)(U^{\mu^*})^T U^\mu$. From the triangular inequality of the Euclidean distance among the relevant vectors, one has

$$|N-2D_\mu| = N\,|\alpha_\mu| \leq N\,|\alpha_{\mu^*\mu}| + 2D_{\mu^*} \qquad (24)$$

So

$$\begin{aligned}
a\ \geq\ & \frac{\lambda}{N}\left((N-2D_{\mu^*}) - \sum_{\mu\neq\mu^*}^{p}[N\,|\alpha_{\mu^*\mu}| + 2D_{\mu^*}]\right) (25) \\
& -\frac{(1-\lambda)}{N}\left(\sum_{\mu=1}^{p}[N\,|\alpha_{\mu^*\mu}| + 2D_{\mu^*}]\right) \\
=\ & \frac{1}{N}\left(2\lambda N - \sum_{\mu\neq\mu^*}^{p}(N\,|\alpha_{\mu^*\mu}| - 2pD_{\mu^*} - N)\right) \geq 0
\end{aligned}$$

The last inequality in (25) is satisfied whenever

$$D_{\mu^*} \leq \frac{N}{2p}\left(2\lambda - 1 - \sum_{\mu\neq\mu^*}^{p}|\alpha_{\mu^*\mu}|\right) \qquad (26)$$

Case 2: Assume Z is closed to a stored pattern U^{μ^*} where μ^* is an even number. Then flow at Z will point to U^{μ^*+1} if

$$[T(t)Z]_k u_k^{\mu^*+1} \geq 0, \forall\ k.$$

From (22), one can obtain the same conclusion as in (26).

B. If $\lambda \to 0$

Case 1: Assume Z is closed to a stored pattern U^{μ^*} where μ^* is an odd number. Then flow at Z will point to U^{μ^*+1} if

$$[T(t)Z]_k u_k^{\mu^*+1} \geq 0, \forall\ k.$$

Case 2: Assume Z is closed to a stored pattern U^{μ^*} where μ^* is an even number. Then flow at Z will point to U^{μ^*} if

$$[T(t)Z]_k u_k^{\mu^*} \geq 0, \forall\ k.$$

Similar analysis has been performed for the above two cases, the obtained result is

$$D_{\mu^*} \leq \frac{N}{2p}\left(1 - 2\lambda - \sum_{\mu\neq\mu^*}^{p}|\alpha_{\mu^*\mu}|\right). \qquad (28)$$

Finally, combining (26) with (28) yields (7). The proof is thus completed.

Appendix II

Proof of theorem 2:

Case 1: Assume the network reaches a stored pattern $U^{\mu^*}$where $\mu^* \in ODD$. Its state will not change as long as

$$sgn[T(t)U^{\mu^*}] = U^{\mu^*} \qquad (29)$$

Condition (29) can be rewritten as

$$[T(t)U^{\mu^*}]_k u_k^{\mu^*} > 0,\ \forall\ k. \qquad (30)$$

If $U^{\mu^*}$is an odd pattern, one can find the lower bound of λ (i.e., $\lambda > \ \eta_k(\mu^*)$) which ensures (30). Substituting $Z = U^{\mu^*}$into (23) and let $D_{\mu^*} = 0$, one has $[T(t)U^{\mu^*}]_k u_k^{\mu^*} =$

$$\begin{aligned}
& \frac{\lambda}{N}\left(N + \sum_{\mu\in ODD,\mu\neq\mu^*}(N-2D_{\mu^*\mu})u_k^\mu u_k^{\mu^*}\right. \\
& \left. + \sum_{\mu\in EVEN}(N-2D_{\mu^*\mu})u_k^{\mu+1}u_k^{\mu^*}\right) \\
& + \frac{(1-\lambda)}{N}\left(\sum_{\mu\in EVEN}(N-2D_{\mu^*\mu})u_k^\mu u_k^{\mu^*}\right. \\
& \left. + \sum_{\mu\in ODD}(N-2D_{\mu^*\mu})u_k^{\mu+1}u_k^{\mu^*}\right) \\
=\ & \lambda\sum_{\mu=1}^{p}(-1)^\mu(u_k^{\mu+1} - u_k^\mu)u_k^{\mu^*}\alpha_{\mu^*\mu} \\
& + \sum_{\mu\in ODD} u_k^{\mu+1}u_k^{\mu^*}\alpha_{\mu^*\mu} \\
>\ & 0
\end{aligned}$$

The last inequality is satisfied whenever (10) hold. Moreover, since $\lambda > \eta_k(\mu^*)$ must hold for $k = 1, ..., N$. One must choose

$$\lambda > \eta(\mu^*) = \max_k\{\eta_k(\mu^*)\}$$

which is the $\mu^* \in ODD$ case in (9).

Case 2: Assume the network reaches a stored pattern $U^{\mu^*}$where $\mu^* \in EVEN$. Substituting $Z = U^{\mu^*}$into (26) and let $D_{\mu^*} = 0$, similar analysis has been performed and the result is $[T(t)U^{\mu^*}]_k u_k^{\mu^*}\ > 0$ whenever $\lambda <$

$\eta_k(\mu^*)$. Again, since $\lambda < \eta_k(\mu^*)$ must hold for $k = 1, ..., N$. One must choose

$$\lambda < \eta(\mu^*) = \min_k \{\eta_k(\mu^*)\}$$

which is the $\mu^* \in EVEN$ case in (9). The proof is thus completed.

References

[1] S. Amari, "Learning patterns and pattern sequences by self-organizing nets of threshold elements," *IEEE Trans. Computers,"* vol. 21, no.11, pp. 1197-1206, Nov. 1972.

[2] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," in *Proc. Nat. Acad. Sci.* USA, vol. 81, pp. 3088-3092, 1984.

[3] W. J. Wang and D. L. Lee, "A modified bidirectional decoding strategy based on the BAM structure,"*IEEE Trans. Neural Networks,* vol. 4, no. 4, pp. 710-717, July 1993.

[4] D. L. Lee and W. J. Wang, "Neighbor-layer updating in MBDS for the recall of pure bipolar patterns in gray-scale noise,"*IEEE Trans. Neural Networks,* vol. 6, no. 6, pp. 1478-1489, Nov. 1995.

[5] D. L. Lee and W. J. Wang, "A multivalued bidirectional associative memory operating on a complex domain," *Neural Networks*, vol. 11, pp. 1623-1635, 1998.

[6] D. L. Lee, "A discrete sequential bidirectional associative memory for multistep pattern recognition," *Pattern Recognition Letters,* vol. 19, pp. 1087-1102, 1998.

[7] D. L. Lee, "New stability conditions for Hopfield networks in partial simultaneous update mode,"*IEEE Trans. Neural Networks,* vol. 10, no. 4, pp. 975-978. , July. 1999.

[8] M. Morita, "Memory and learning of sequential patterns by nonmonotone neural networks," *Neural Networks,* vol. 9, no. 8, pp. 1477-1489, 1996.

[9] H. Sompolinsky and I. Kanter, "Temporal association in asymmetric neural networks," *Physical. Review. Letters,* vol. 57, no. 22, Dec. 1986.

[10] D. Kleinfeld, "Sequential state generation by model neural networks," in *Proc. Nat. Acad. Sci.* USA, vol. 83, pp. 9469-9473, 1986.

[11] I. Guyon, L. Personnaz, J. P. Nadal, and G. Dreyfus, "Storage and retrieval of complex sequences in neural networks," *Physical Review A,* vol. 38, no. 12, pp. 6365-6372. Dec. 1988.

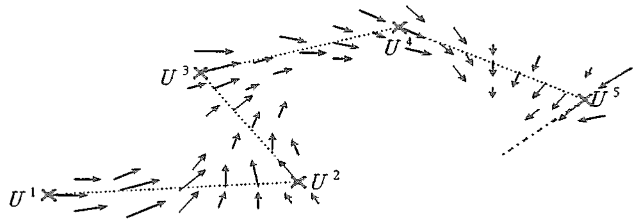

Figure 1 – The conventional flow vector field distribution in the state space

[12] A. N. Michel, and J. A. Farrell, "Associative memories via artificial neural networks," *IEEE Control Systems Magazine*, vol. 10, pp. 6-17, Apr. 1990.

[13] L. Personnaz, I. Guyon, and G. Dreyfus, "Collective computational properties of neural networks: New learning mechanisms," *Physical Review A,* vol. 34, no. 5, pp. 4217-4228. Nov. 1986.

[14] R. W. Zhou, and C. Quek, "DCBAM: A discrete chainable bidirectional associative memory," *Pattern Recognition Letters,* vol. 17, pp. 985-999, 1996.

[15] M. L. Minsky and S. Papert, *Perceptrons,* MIT press, Cambridge, MA, 1969.

[16] S. I. Gallent, "Optimal linear discriminants," in *Proc. Eighth Int. Conf. Patt. Recog.*, Paris, France, pp. 849-852, 1986.

[17] Y. C. Ho, and R. Kashyap, "An algorithm for linear inequalities and its applications," *IEEE Trans. Electron. Comput.* vol. 14, pp. 683-688, 1965.

[18] W. Krauth and M. Mezard, "Learning algorithms with optimal stability in neural networks," *J. Phys. A: Math. Gen.*, vol. 20, pp. L745-752, 1987.

[19] A. Albert, *Regression and the Moore-Penrose Pseudoinverse,* Academic, New York, 1972.

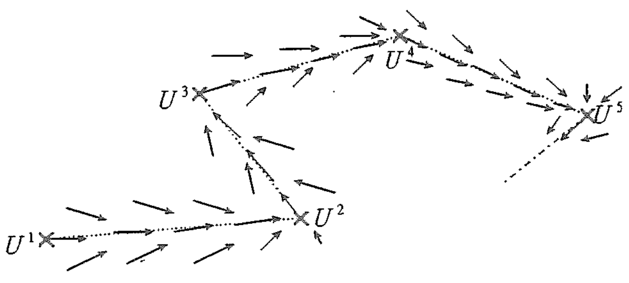

Figure 2 – The desired flow vector field distribution in the state space.

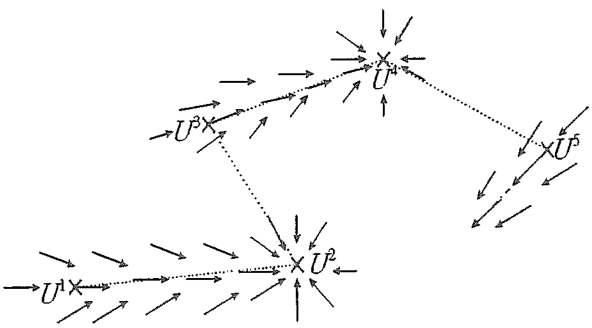

Figure 3 – The actual flow vector field distribution in the state space. Pattern sequence $U^1 \rightarrow U^2 \rightarrow ... \rightarrow U^p$ is stored according to (5) with $\lambda(t) \rightarrow 1$

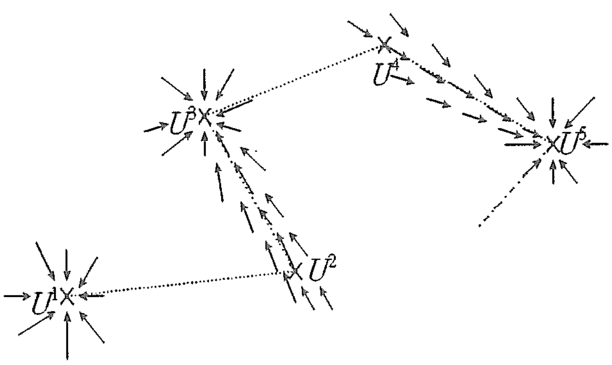

Figure 4 – $\lambda(t) \rightarrow 0$

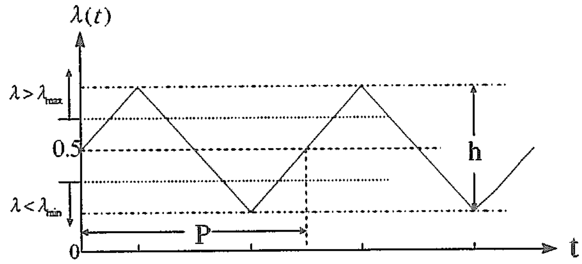

Figure 5 – The waveform of $\lambda(t)$. A triangular wave with period P and height h

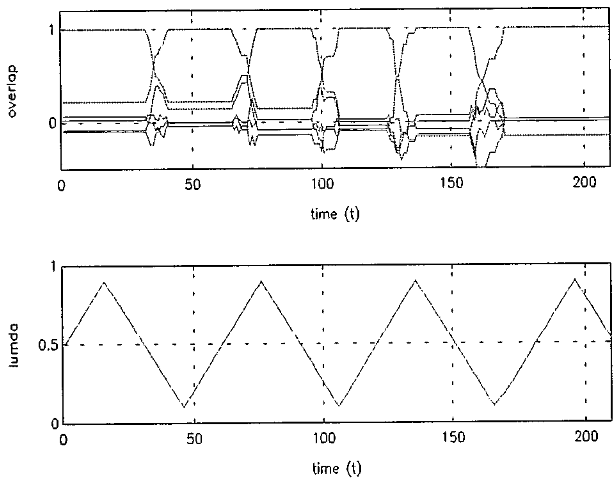

Figure 6 – Examples of the TVHMs recalling procedure (see (13)). The stored pattern sequence$U^1 \rightarrow ... \rightarrow U^p \longleftrightarrow U^p$ is generated by random, $p = 6, N = 100$, $\tau = 10$, $h = 0.8$, and $\Delta t = 1$. $P = 6\tau$,

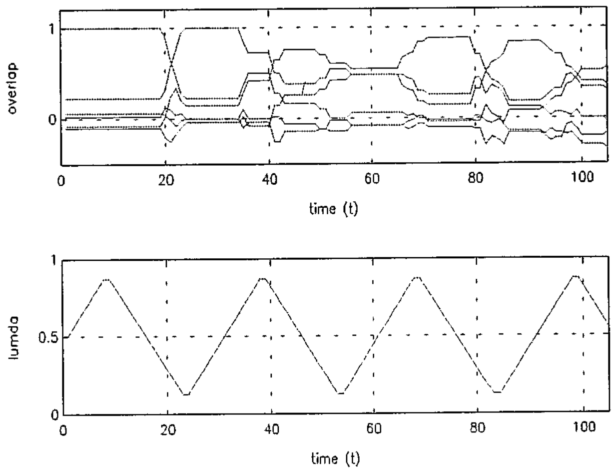

Figure 7 – $P = 3\tau$,

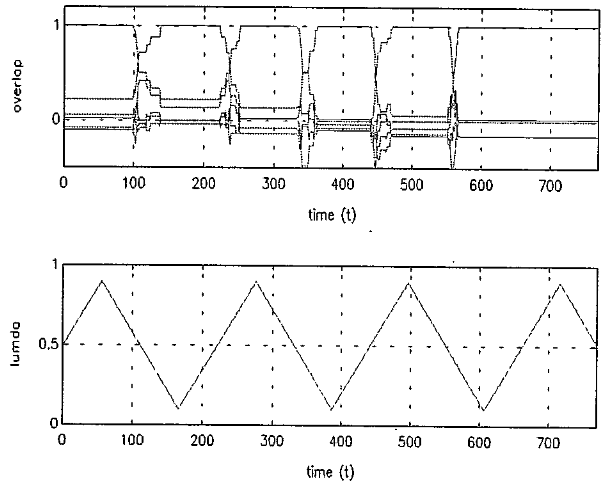

Figure 8 – $P = 22\tau$

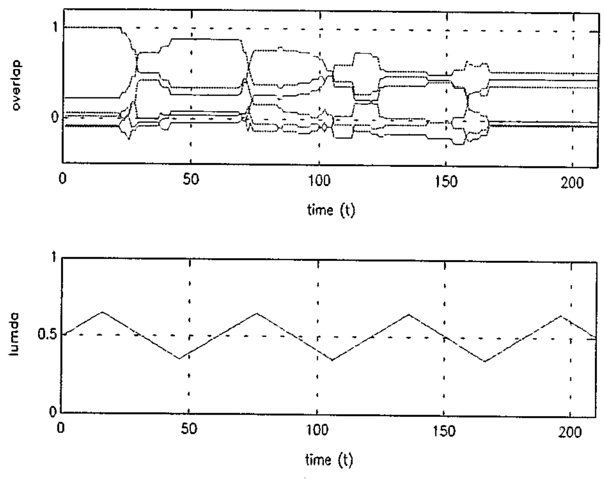

Figure 9 – Imperfect recall of a pattern sequence due to an improper value of h ($h = 0.3 > 2(0.5 - \lambda_{\min}) = 0.2882$)

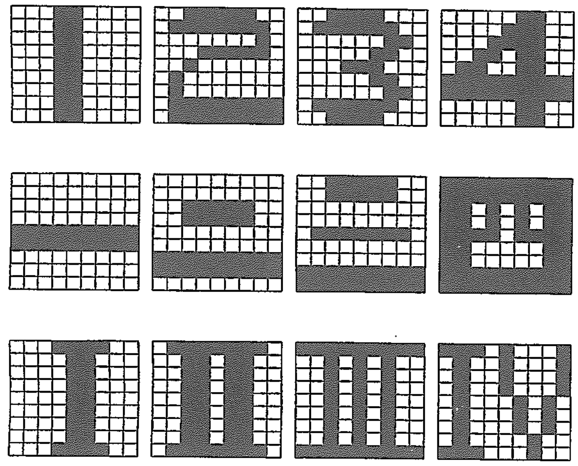

Figure 10 – Three pattern sequences for the pattern sequence recognition example

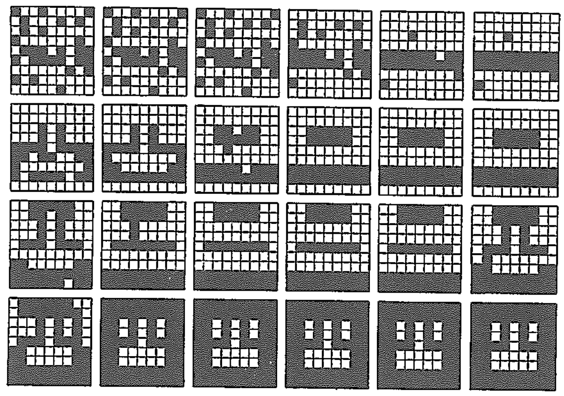

Figure 11 – Convergence of the initial state $Z(0) = \tilde{U}^5$(upper-left) to the final state $Z(92) = U^8$(lower-right). The initial state $Z(0)$ is obtained from inverting ten pixels (about 25 %) of U^5

TABLE I

RECALL PROBABILITY AGAINST HAMMING DISTANCE FOR PTVHMS

$H(\tilde{U}^\mu, U^\mu)$	2	4	6	8	10	12	14	16	18	20	22	24
$\tilde{U}^1 \rightarrow U^4$	1	1	1	1	1	0.99	0.97	0.89	0.86	0.76	0.63	0.55
$\tilde{U}^5 \rightarrow U^8$	0.91	0.82	0.78	0.77	0.76	0.76	0.73	0.70	0.65	0.65	0.54	0.47
$\tilde{U}^9 \rightarrow U^{12}$	1	0.98	0.90	0.83	0.79	0.75	0.71	0.59	0.57	0.48	0.35	0.27

Figure 12 –