

# A Chaotic Memory Search Model Based on Associative Dynamics Using Features in Stored Patterns

K. Kobayashi<sup>1†</sup>, K. Watanabe<sup>2</sup>, and M. Obayashi<sup>1</sup>

<sup>1</sup> Yamaguchi University, 2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan

<sup>2</sup> Hitachi Software Engineering Company, 6-81, Onoe-cho, Naka-ku, Yokohama 231-0015, Japan

† k@nn.csse.yamaguchi-u.ac.jp

**Abstract:** A new chaotic memory search model based on associative dynamics using features in stored patterns is proposed. In the present paper, two kinds of features are considered; external and internal ones. The former is assigned by a designer and the latter is automatically assigned by competitive learning. The control of chaotic and static states is realized using a presynaptic inhibition model. The performance of the proposed model is evaluated through computer simulation.

**Keywords:** Chaotic neural network, Memory search, Feature, Associative dynamics

## 1. Introduction

Memory search is one of interesting human information processing mechanisms. Aihara et al. proposed a chaotic neuron model<sup>1)</sup> and the associative dynamics in a chaotic neural network (CNN) was analyzed<sup>2)</sup>. The network could chaotically, i.e. aperiodically retrieve stored patterns. Since the trajectory of the network states covers almost whole region in a phase space, a target pattern could be retrieved efficiently.

Nara et al. proposed a memory search model using chaotic dynamics<sup>3)</sup>. In their model, the number of connection weights is adjustable so as to control between chaotic and static states.

Deguchi et al. proposed a memory search model using presynaptic inhibition<sup>4,5)</sup>. In their model, memory search is realized based on association by assigned features. The features, however, are not used directly and efficiently. Since backpropagation learning is employed to extract the features, computational cost is heavy. Furthermore, all the features are assigned to stored patterns by a designer in advance. Therefore, the assignment of features highly depends on knowledge of the designer<sup>6)</sup>.

In the present paper, a new chaotic memory search model, which treats two kinds of features, external and internal ones, is proposed. The external feature is assigned by a designer but the internal feature is automatically assigned by competitive learning. Through computer simulation, the performance of the proposed model is evaluated.

## 2. Chaotic neural network

The dynamics of CNN is described as follows<sup>1)</sup>:

$$x_i(t+1) = f(y_i(t+1) + z_i(t+1)), \quad (1)$$

$$y_i(t+1) = k_r y_i(t) - \alpha x_i(t) + a_i, \quad (2)$$

$$z_i(t+1) = k_f z_i(t) + \sum_{j=1}^n w_{ij} x_j(t), \quad (3)$$

where  $x_i$  is the output of the  $i$ th neuron,  $y_i$  and  $z_i$  are internal states of refractoriness and feedback inputs of the  $i$ th neuron respectively,  $k_r$  and  $k_f$  are decay parameters for  $y_i$  and  $z_i$  respectively,  $\alpha$  is the scaling parameter,  $a_i$  is the external input of the  $i$ th neuron,  $n$  is the number of neurons,  $w_{ij}$  is the connection weight from the  $j$ th neuron to the  $i$ th one,  $f(\cdot)$  is the sigmoidal output function:

$$f(x) = \frac{1 - \exp(-x/\epsilon)}{1 + \exp(-x/\epsilon)}, \quad (4)$$

where  $\epsilon$  is the slope of sigmoid function.

If the values of  $y_i$ 's are relatively larger than those of  $z_i$ 's, the network could take a chaotic state and produce the series of the stored patterns aperiodically.

## 3. Memory search

Chaotic dynamics is useful to search a stored pattern efficiently embedded in the pattern space. The stored pattern is regarded as a point in the phase space. Therefore, it is defined that memory search is to find the point in the phase space using a key information.

The larger the pattern space and the number of patterns to be stored are, the more difficult memory search is. Chaotic dynamics, however, is effective to transit states around a particular region in the phase space.<sup>7,8,9)</sup>

The conventional model proposed by Deguchi et al.<sup>4,5)</sup> is described (Section 3.1). Then a new memory search model is proposed (Section 3.2).

### 3.1 Conventional model

Deguchi et al. proposed a memory search model to control between chaotic and static states by adjusting the balance between refractoriness and feedback inputs. More specifically, presynaptic inhibition is modeled<sup>4,5)</sup>.

In the conventional model, each feature is assigned to the corresponding pattern. The correspondence between the feature and pattern is determined by backpropagation learning.

At first, a set of features for a target pattern is given to the network. A stored pattern is dynamically retrieved in the network. On the other hand, a set of features corresponding to a stored pattern is extracted. If both sets of features, i.e. feature inputs and extracted features are identical, it is regarded that the correct pattern corresponding to the given set of features is retrieved. Then, the state of the network is switched from dynamic to static by controlling the intensity of presynaptic inhibition. The control of presynaptic inhibition is explicitly performed using the following equation instead of Eq.(3).

$$z_i(t+1) = k_f z_i(t) + \gamma(s) \sum_{j=1}^n w_{ij} x_j(t), \quad (5)$$

where  $\gamma(s)$  denotes the function of presynaptic inhibition:

$$\gamma(s) = \begin{cases} 1.0 & (s < s_{th}) \\ \gamma_c & (s \geq s_{th}) \end{cases}, \quad (6)$$

where  $s_{th}$  is the threshold, which controls between chaotic and static states and  $\gamma_c$  is constant satisfying  $0 < \gamma_c < 1$ .

The algorithm of the conventional model is summarized as follows:

1. The backpropagation network is trained to correspond each set of features to the stored pattern.
2. A set of features for the desired stored pattern,  $x_i^F$ , is given.
3. A stored pattern is dynamically retrieved in the network.
4. The set of features,  $x'_i(t)$ , is calculated using the retrieved pattern through the backpropagation network.
5. The similarity between  $x'_i(t)$  and  $x_i^F$  is calculated using the following equation.

$$s = \frac{1}{2n} \sum_{i=1}^{F_n} |x_i^F - x'_i(t)|, \quad (7)$$

where  $F_n$  is the number of the sets of features.

6. If  $s < s_{th}$  then  $\gamma(s) = 1.0$ . Otherwise, go back to step 3.

### 3.2 Proposed model

In the conventional model, features are only used for comparison between input and output features and not used directly. The feature extraction network is trained by back-propagation learning. Therefore, it would be time consuming and the size of the network tends to be large. And also the assignment of features highly depends on knowledge of the designer.

In the present paper, two kinds of features, the external and internal features, are considered. The external feature represents the inherent feature of stored patterns and internal feature depends on the designer. For example, let consider human face, the external features are nationality, gender and

occupation and the internal features are small face, big eyes and so on. Then, these features are directly used for memory search<sup>10)</sup>. As a result, the stored pattern could be searched more effectively.

**Structure of proposed model** The proposed model is shown in Fig.1. It consists of four layers; external feature (EF) layer, internal feature (IF) layer, chaotic neural network (CNN) layer and feature extraction (FE) layer. The EF layer has the connections to CNN and IF layers. The IF layer has the connection to CNN layer. Then, CNN layer has the connection to FE layer.

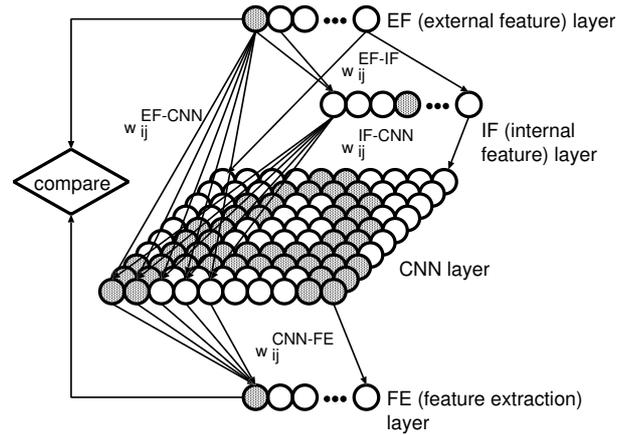


Figure 1: A Proposed model

The internal features are acquired by competitive learning to overcome the dependency on the designer. In the proposed model, the following dynamics is used instead of Eq.(5).

$$z_i(t+1) = k_f z_i(t) + \gamma(s) \sum_{j=1}^n w_{ij}^{CNN} x_j(t) + \sum_{j=1}^{f^{EF}} w_{ij}^{EF-CNN} x_j^{EF} + \sum_{j=1}^{f^{IF}} w_{ij}^{IF-CNN} x_j^{IF}, \quad (8)$$

where  $\gamma(s)$  denotes the presynaptic inhibition defined in Eq.(6),  $f^{EF}$  and  $f^{IF}$  are external and internal features respectively,  $x_j^{EF}$  and  $x_j^{IF}$  are the outputs of the  $j$ th neuron in EF and IF layers respectively,  $w_{ij}^{CNN}$  is the connection weight in CNN layer and  $w_{ij}^{EF-CNN}$  and  $w_{ij}^{IF-CNN}$  are the connection weights between CNN and EF and those between CNN and IF layers respectively:

$$w_{ij}^{CNN} = \sum_{k=1}^P x_i^{CNN}(k) x_j^{CNN}(k), \quad (9)$$

$$w_{ij}^{EF-CNN} = \sum_{k=1}^P x_i^{EF}(k) x_j^{CNN}(k), \quad (10)$$

$$w_{ij}^{IF-CNN} = \sum_{k=1}^P x_i^{IF}(k) x_j^{CNN}(k), \quad (11)$$

where  $x_i^{CNN}(k)$  represents the  $i$ th element for the  $k$ th stored pattern,  $x_i^{EF}(k)$  and  $x_i^{IF}(k)$  are the  $i$ th element of internal and external features for the  $k$ th stored pattern, respectively, and  $P$  is the number of stored patterns.

In Eq.(8), the third and fourth terms are crucial for searching stored patterns.

The connection weights,  $w_{ij}^{EF-IF}$  and  $w_{ij}^{CNN-FE}$ , are given as follows:

$$w_{ij}^{EF-IF} = \sum_{k=1}^P x_i^{EF}(k)x_j^{IF}(k), \quad (12)$$

$$w_{ij}^{CNN-FE} = \sum_{k=1}^P x_i^{CNN}(k)x_j^{EF}(k). \quad (13)$$

**Assignment of internal features** In the present paper, the external features are assigned by the designer in advance. On the other hand, the internal features are assigned by competitive learning described below <sup>11)</sup>.

1. The connection  $w_{ji}$  between the  $i$ th unit in the input layer and the  $j$ th unit in the competitive layer are randomly initialized in interval  $[0, 1]$  with the following constraint.

$$\sum_i w_{ji} = 1 \quad (14)$$

2. The internal state  $S_j$  of the competitive unit is calculated as follows:

$$S_j = \sum_i w_{ji}x_i, \quad (15)$$

where  $x_i \in \{0, 1\}$  is the  $i$ th element of the stored pattern.

3. A unit with the largest value of  $S_j$ 's is selected as a winner and its connections are updated using the following equation.

$$\Delta w_{ji} = g \left( \frac{x_i}{m} - w_{ji} \right), \quad (16)$$

where  $g$  denotes the learning coefficient and  $m$  is the value which satisfies  $\sum_i x_i/m = 1$ .

4. Go back to step 2 until the network is converged.

In the present paper, competitive units are added dynamically. Namely, the network initially has only one unit. A new pattern is presented to the network and then the distance with each pattern presented so far is calculated. If the distance is larger than a threshold then a new unit is added and its connections are initialized as Eq.(14).

Using the above algorithm, all the stored patterns are categorized. The internal features are determined by those categories. For example, let consider four patterns; A, B, C and D. These are categorized as shown in Table 1.

As seen in Table 1, patterns C and D belong the same category and therefore the internal features of C and D are supposed to be identical. After that, internal features are determined according to the above categorization shown in Table 2.

Table 1: An example of categories

Pattern	A	B	C	D
Category	1	2	3	3

Table 2: An example of internal features

Internal feature	A	B	C	D
1	+1	-1	-1	-1
2	-1	+1	-1	-1
3	-1	-1	+1	+1

In this table, there are three internal features because the number of categories is three. One element of internal features corresponding to the category number has +1 and the other elements are -1's.

**Algorithm of memory search** The proposed algorithm is described below.

1. The stored patterns are presented to the competitive network and then are categorized. After that, the internal features are assigned.
2. The connections  $w_{ij}^{CNN}$ ,  $w_{ij}^{EF-CNN}$ ,  $w_{ij}^{EF-IF}$ ,  $w_{ij}^{IF-CNN}$  and  $w_{ij}^{CNN-FE}$  are calculated.
3. The target pattern is presented to produce the outputs in EF layer.
4. The outputs are propagated through IF layer and then  $x_i^{IF}$  is calculated using the following equation.

$$x_i^{IF} = \begin{cases} -1.0 & \left( \sum_{j=1}^{fEF} w_{ij}^{EF-IF} x_j^{EF} < 0 \right) \\ 0 & \left( \sum_{j=1}^{fEF} w_{ij}^{EF-IF} x_j^{EF} = 0 \right) \\ +1.0 & \left( \sum_{j=1}^{fEF} w_{ij}^{EF-IF} x_j^{EF} > 0 \right) \end{cases} \quad (17)$$

5. The stored pattern is retrieved in CNN layer through EF and IF layers (Eq.(8)).
6. The set of features,  $x_i^{FE}(t)$ , is produced in FE layer through the pattern in CNN layer.

$$x_i^{FE}(t) = f \left( \sum_{j=1}^{fEF} w_{ij}^{CNN-FE} x_j(t) \right), \quad (18)$$

where  $f(\cdot)$  is the sigmoid function:

$$f(x) = \frac{1 - \exp(-x/\epsilon')}{1 + \exp(-x/\epsilon')}, \quad (19)$$

where  $\epsilon'$  is the slope of sigmoid function in FE layer.

7. The similarity between  $x_i^{FE}(t)$  and  $x_i^{EF}$  is calculated using the following equation:

$$s = \frac{1}{2n} \sum_{i=1}^{fEF} |x_i^{EF} - x_i^{FE}(t)|. \quad (20)$$

8. If  $s < s_{th}$  then  $\gamma(s) = 1.0$ . Otherwise, go back to step 5.

## 4. Computer simulation

The performance of the proposed model is compared with that of the conventional model through computer simulation.

### 4.1 Environment

The values of the parameters used in the simulation are shown in Table 3.

Table 3: Parameter

Parameter	Conventional model	Proposed model
$k_r$	0.5	0.5
$k_f$	0.55	0.55
$\alpha$	10.0	10.0
$a_i$	2.0	2.0
$\epsilon$	0.015	0.015
$\epsilon'$	—	5.0
$s_{th}$	0.45	0.2
$\gamma_c$	0.4	0.4

In the proposed model, the number of external features  $f^{EF} = 3$  and the third elements of external features are set to  $-1$  to prevent retrieving the reverse patterns.

Four alphabet patterns shown in Fig.2, J, D, A and H, are used for patterns to be stored and their external features are assigned according to whether lateral symmetry or not and vertical symmetry or not as shown in Table 4.

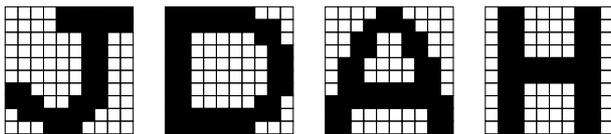


Figure 2: Stored patterns

Table 4: Assignment of external features

External feature	J	D	A	H
Lateral symmetry	-1	-1	+1	+1
Vertical symmetry	-1	+1	-1	+1
Dummy	-1	-1	-1	-1

These four patterns are presented to the competitive network. As a result, the internal features shown in Table 5 are obtained. In the simulation, the threshold of distance is set as  $d_{th} = 0.4$ .

Table 5: Assignment of internal features

Internal feature	J	D	A	H
1	+1	-1	-1	-1
2	-1	+1	-1	-1
3	-1	-1	+1	-1
4	-1	-1	-1	+1

### 4.2 Speed of memory search

Figure 3 to 6 shows the averaged number of steps to find a stored pattern in 100 trials. In this figure, the abscissa represents the number of steps and the ordinate is successful rate of memory search. As seen in these figures, the speed of the proposed model is dramatically faster than that of the conventional model.

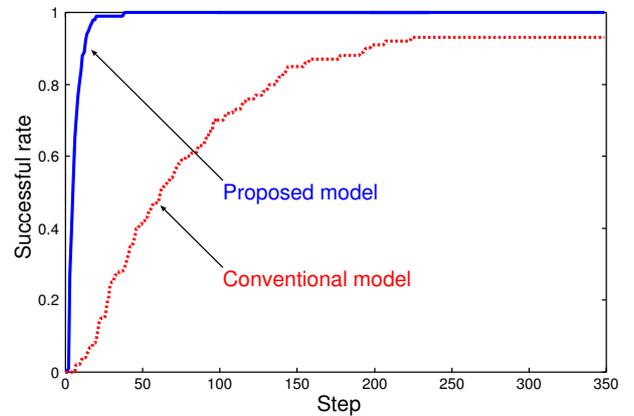


Figure 3: Successful rate for target pattern J

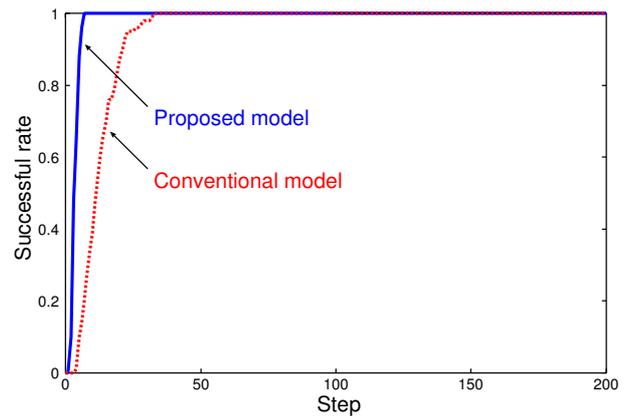


Figure 4: Successful rate for target pattern D

### 4.3 One-to-many correspondence

One-to-many correspondence refers to one feature pattern corresponds to plural stored patterns. In memory search, all

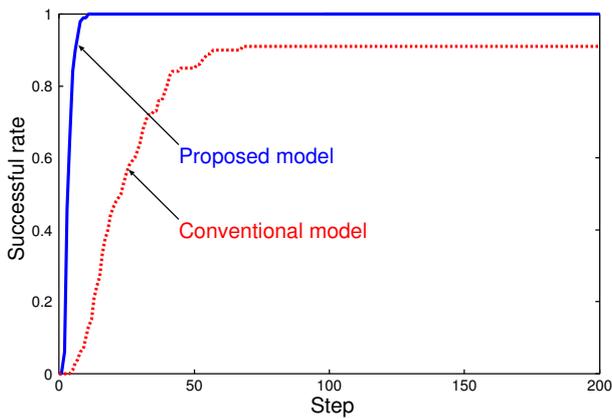


Figure 5: Successful rate for target pattern A

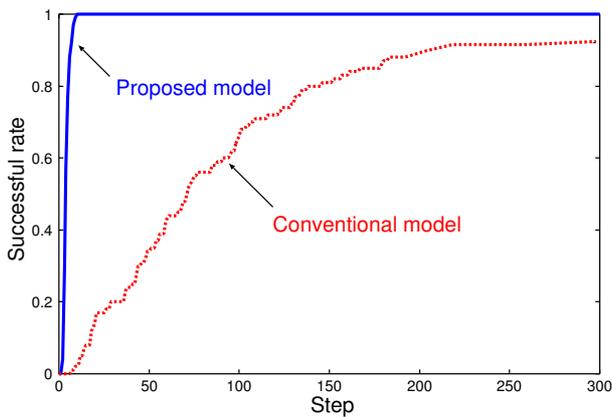


Figure 6: Successful rate for target pattern H

the features of a target pattern are not always given. For example, if the first element of external features is known as  $-1$  and the second element is unknown, i.e. we know partial information about lateral asymmetry, the target pattern is J or D in this case. In the present paper, the unknown feature is denoted as 0. Therefore, one-to-many correspondence is realized using the feature vector  $(-1, 0)$ .

The simulation results for one-to-many correspondence are shown in Table 6 and Table 7. In these tables, the number of successful trials for searching stored patterns and the number of failed trials are shown in 100 trials. The failure means that memory search is not succeeded in 1000 steps. As seen in these tables, it is obvious that each pattern has different size of attractor because each number of successful trials is different. The number of successful trials for each pattern in the proposed model is fairly balanced than the conventional model.

## 5. Discussion

### 5.1 Effect of the number of internal features

In the simulation, we used four internal features. Each neuron in IF layer corresponds to one and only one stored pat-

Table 6: One-to-many correspondence for conventional model

Input	J	D	A	H	Failure
Lateral asymmetry	6	94	0	0	0
Vertical asymmetry	24	0	65	0	11
Lateral symmetry	0	0	76	19	5
Vertical symmetry	0	95	0	5	0

Table 7: One-to-many correspondence for proposed model

Input	J	D	A	H	Failure
Lateral asymmetry	11	89	0	0	0
Vertical asymmetry	21	0	79	0	0
Lateral symmetry	0	0	54	46	0
Vertical symmetry	0	77	0	23	0

tern. This means that recognition cells are formed. The number of the elements of features, however, increases as the number of the stored patterns increases.

The result for different IF, the number of internal features, is shown in Fig.7. In this figure, the target pattern is assumed as H. As shown in this figure, the performance of memory search is improved as IF increases. At the same time, the speed of memory search in the proposed model is much faster than the conventional model even if  $IF=0$  because there are connections between EF and CNN layers.

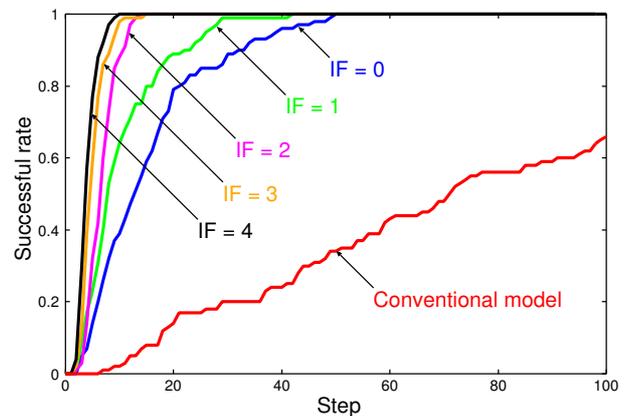


Figure 7: Results of memory search for different number of internal features

### 5.2 Trajectory of memory search

Through computer simulation, it is shown that the proposed model could search stored patterns much faster than the conventional model. This is because the state of the network changes to the direction of feature. The outputs of FE layer

are plotted for target pattern H in two dimensional plane in Fig.8 and 9. In these figures, point  $(+1, +1)$  refers to target pattern H and  $(-1, -1)$ ,  $(-1, +1)$  and  $(+1, -1)$  are pattern J, D and A, respectively. As seen in these figures, the proposed model could search all over the stored patterns. The trajectory concentrates around pattern D because the attractor basin of D is large. Therefore, the conventional model highly depends on the attractor basin.

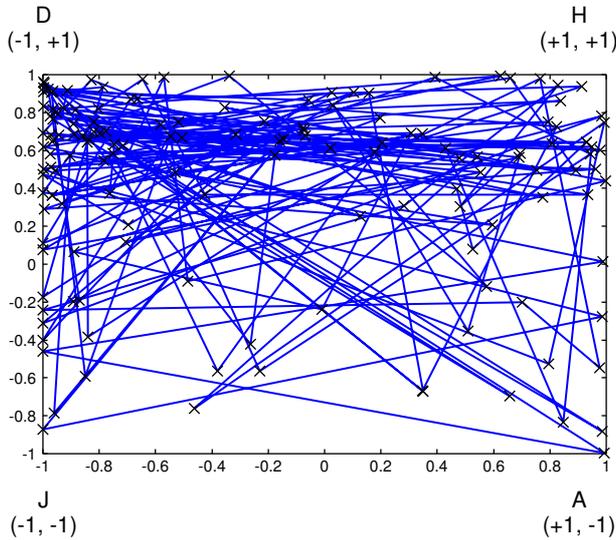


Figure 8: Search trajectory in the conventional model

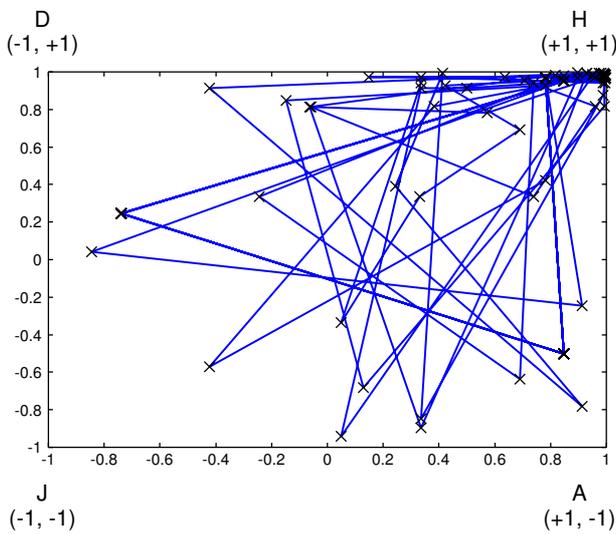


Figure 9: Search trajectory in the proposed model

## 6. Summary and conclusion

The present paper has proposed a new chaotic memory search model based on association by features. The internal features are extracted using competitive learning so that the assignment does not depend on the designer. Through

computer simulation, it is clarified that speed of memory search is dramatically improved compared with the conventional model.

In the future, we need to investigate the performance of our model when the number of stored patterns increases and correlation between patterns is strong. A memory search model incorporating nonmonotone output function<sup>12,13)</sup> is promising in such situation.

## References

- [1] Aihara, K., Takabe, T. and Toyoda, M.: "Chaotic neural networks", *Physics Letter A*, Vol.144, No.6,7, pp.333–340, 1990.
- [2] Adachi, M. and Aihara, K.: "Associative dynamics in a chaotic neural network", *Neural Networks*, Vol.10, No.1, pp.83–98, 1997.
- [3] Nara, S. and Davis, P.: "Chaotic wandering and search in a cycle-memory neural network", *Progress of Theoretical Physics*, Vol.88, No.5, pp.845–855, 1992.
- [4] Deguchi, T. and Ishii, N.: "Control of associative dynamics by matching features in chaotic neural network", *Trans. on IEICE*, Vol.J78-D-II, No.8, pp.1223–1230, 1995 (in Japanese).
- [5] Deguchi, T. and Ishii, N.: "Model of presynaptic inhibition for search access", *Trans. on IEICE*, Vol.J80-D-II, No.2, pp.618–625, 1997 (in Japanese).
- [6] Iwasaki, M., Hashiyama, T. and Okuma, S.: "Self-organizing feature extraction using associative memory", *Trans. on IEE Japan*, Vol.120, No.10, pp.1467–1474, 2000 (in Japanese).
- [7] Ikeda, K., Otsuka, K. and Matsumoto, K.: "Maxwell-block turbulence", *Progress of Theoretical Physics Supplement*, No.99, pp.295–324, 1989.
- [8] Kaneko, K.: "Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements", *Physica*, Vol.41D, pp.137–172, 1990.
- [9] Tsuda, I.: "Chaotic itinerancy as a dynamical basis of hermeneutics in brain and mind", *World Futures*, Vol.32, pp.167–184, 1991.
- [10] Watanabe, K., Obayashi, M. and Kobayashi, K.: "Memory search using chaotic neural network with feature dynamics", *Proc. of ICONIP2000 (Korea)*, pp.1124–1129, 2000.
- [11] Dayhoff, J. E.: "Neural network Architectures: An introduction", Van Nostrand Reinhold, 1990.
- [12] Morita, M.: "Associative memory with nonmonotone dynamics", *Neural Networks*, Vol.6, No.1, pp.115–126, 1993.
- [13] Nakagawa, M.: "A super memory retrieval with chaos associative model", *Journal of the Physical Society of Japan*, Vol.68, No.7, pp.2457–2465, 1999.