The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Chaotic Dynamical Associative Memory Model Using Supervised Learning

Masanao Obayashi, Yoshie Yano, Kunikazu Kobayasi, Takashi Kuremoto

Graduate School of Science and Engineering Yamaguchi University, 2-16-1 Tokiwadai Ube Yamaguchi Japan
(Tel : 0836-85-9518; Fax : 0836-85-9501)
{obayashi, yano, k, wu}@nn.csse.yamaguchi-u.ac.jp

Abstract: While chaotic neural networks proposed by Aihara has some problems that storage capacity is too small and success judgment of association is difficult, the model using bidirectional associative memory (BAM) that proposed by Chartier is learned using supervised learning, is high storage capacity and high-speed recall ability. In this paper, we changed Chartier's et al's evaluation measure to minimize, and modified learning rule together. As a result, the obtained learning rule has recall ability better than Chartier's, and has learning speed for connection weight faster than Chartier's. Additionally, it can also treat multi-valued image. However, it can't recall chaotically as CNN, because used autoassociative memory model adopts output function which makes networks stable states. So, in order to accomplish chaotic behavior, we construct the model that can escape from steady states by generating control signal automatically when network becomes stable states.

Keywords: Chaos Neural Network, Associative Memory, Supervised Learning.

I. INTRODUCTION

A study of associative memory has been actively done for this past 30 years. Since an interconnected network was proposed by Hopfield (Hopfield Network: HN), the study came to be more active. After storing patterns in HN, for any initial patterns, HN converges to a local stable state, that is, it recalls a stored pattern or its similar pattern. However, HN has some problems that it can't escape from local stable states.

Otherwise, because chaotic phenomenon is observed in neural networks of human brain, neuron with chaotic dynamics is modeled and called chaotic neuron. Chaotic neuron model proposed by Aihara has some properties that are spatio-temporal increment of inputs from other plural neurons, refractoriness, and continuous outputs [1,2]. Chaotic neural networks (CNN) constructed with this chaotic neuron model will be able to recall any stored patterns dynamically, if network parameters are set appropriately. Because of this feature, it came to possible to escape from above local stable states. But its storage capacity is too small and HN and Aihara's CNN model can't treat multi-valued image.

Chartier et al proposed the associative memory model of the autoassociative memory and the bidirectional associative memory (BAM) respectively [3], [4,5]. These model have some advantages are high storage capacity and high-speed recall abilities, because connection weight is setting using supervised learning utilizing gradient method. And it can treat not only binary but also multi-valued image.

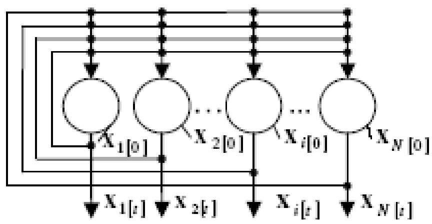

Fig.1. Architecture of the model [3]

In this paper, we changed Chartier's et al's evaluation measure to minimize, and modified learning rule together. As a result, the obtained learning rule has recall ability better than Chartier's, and has learning speed for connection weight faster than Chartier's. Additionally, it can also treat multi-valued image.

However, it can't recall chaotically as CNN, because used autoassociative memory model adopts output function which makes networks stable states. So, in order to accomplish chaotic behavior, we construct the model that can escape from steady states by generating control signal automatically when network becomes stable states.

II. LEARNING RULE

Fig.1 shows the interconnected network architecture used in this paper. $x_{i[0]}$ is an initial value of ith neuron, $x_{i[t]}$ is an output of ith neuron at time t, and N is a number of neurons in the network.

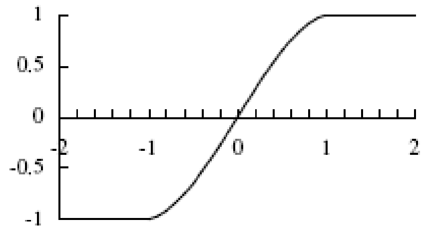

Fig.2. The output function $f(\cdot)$ for $\delta = 0.4$

1. Output Function

The output function in the model is expressed by the following eq. [3].

$$\forall_{i,\dots,N}, \mathbf{x}_{i[t+1]} = f(\mathbf{a}_{i[t]})$$
$$= \begin{cases} 1 & \text{if } \mathbf{a}_{i[t]} > 1, \\ -1 & \text{if } \mathbf{a}_{i[t]} < -1, \\ (\delta+1)\mathbf{a}_{i[t]} - \delta \mathbf{a}_{i[t]}^3 & \text{otherwise,} \end{cases} \quad (1)$$

$\mathbf{a}$ represents the activation vector given by following.

$$\mathbf{a}_{[t]} = \mathbf{W}_{[k]} \mathbf{x}_{[t]}. \quad (2)$$

$\mathbf{W}_{[k]}$ is the connection weight matrix, δ is a general output parameter, and k is the learning trial number (see Eq.(6)). The shape of the output function depends on δ. Fig.2 illustrates the shape of the output function with $\delta = 0.4$.

2. Learning Rule

Chartier's model's learning rule is expressed by Eq.(4) that minimize the error function (3).

$$e = (\mathbf{x}_{[0]} - \mathbf{x}_{[1]})^T (\mathbf{x}_{[0]} - \mathbf{x}_{[1]} - \mathbf{x}_{[2]}), \quad (3)$$

$$\mathbf{W}_{[k+1]} = \mathbf{W}_{[k]} + \eta (\mathbf{x}_{[0]} \mathbf{x}_{[0]}^T - \mathbf{x}_{[t]} \mathbf{x}_{[t]}^T), \quad (4)$$

where η is a general learning parameter, and $\mathbf{x}_{[0]}$ is a stored pattern. But, the error function (3) is unusual. So, we derived learning rule (6) by generally used error function (5) (see Appendix).

$$e = (\mathbf{x}_{[0]} - \mathbf{x}_{[1]})^T (\mathbf{x}_{[0]} - \mathbf{x}_{[1]}), \quad (5)$$

$$\mathbf{W}_{[k+1]} = \mathbf{W}_{[k]} + \eta (\mathbf{x}_{[0]} \mathbf{x}_{[0]}^T - \mathbf{x}_{[1]} \mathbf{x}_{[0]}^T), \quad (6)$$

Eq.(6) is updated by using the autocorrelation of stored patterns and the mutual correlation of stored pattern and outputs used the weight matrices. Learning converges when output is corresponding to stored pattern.

A. Learning of $\mathbf{W}_{[k]}$

Learning is carried according to the following procedure:

1) One pattern is selected from plural stored patterns at random, and setting to $\mathbf{x}_{[0]}$;

2) Compute $\mathbf{x}_{[1]}$ according to the output function (1);

3) Update the weight matrix $\mathbf{W}_{[k]}$ according to Eq.(6);

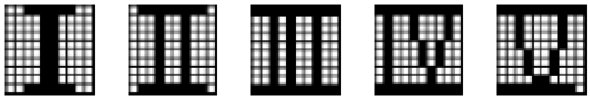

Fig.3. Stored patterns

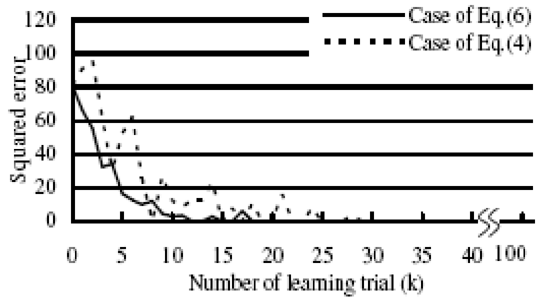

Fig.4. Squared error

Table 1. Associative success rates by different learning rules (%)

	Eq.(4)	Eq.(6)
3 stored patterns	85.0	98.6
4 stored patterns	53.8	98.1

4) Repeat steps 1) to 3) (k in Eq.(6)) until the error function e converges.

3. Computer Simulation

We carried out the association simulation of the model with the changed learning rule. Each parameter is $\delta = 0.1$, $\eta = 0.03$, the stored patterns are 4 patterns of 100 neurons with the value of -1 or 1 shown in Fig.3, and maximum learning number is 100. After learning $\mathbf{W}_{[k]}$, the noisy stored pattern is given as an initial pattern. Fig.4 shows each squared error using Eq.(4),(6). We see that the convergence of Eq.(6) is faster than Eq.(4) in Fig.4. And it is verified associative capacity using Eq.(6) is also superior to using Eq.(4), by simulation using the proposed model explained in Section 3. That is, table 1 shows associative success rate in correlation 0.2 at 3 and 4 stored patterns. Here, associative success rate means the number of associative success per 100 times for one trial, and the average of 10 trials. We see the associative success rate of proposed learning rule (6) is higher than that of (4) from Table 1.

III. PROPOSED MODEL

In this section, we compose the model so that outputs may change when the network becomes a stable state.

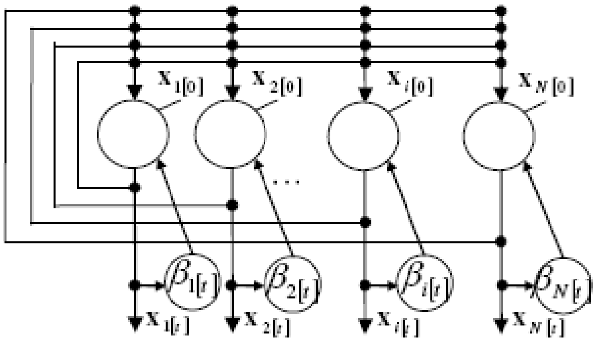

Fig.5. Architecture of the proposed model

Fig.5 shows the architecture of the proposed model. A point different from the previous model is that our model has control neuron β and it output works input to neuron.

1. Improved Output Function

The output function is changed as follows.

$$\forall_{i,\ldots,N}, \mathbf{x}_{i[t+1]} = f\left(\mathbf{a}_{i[t]}\right) = \begin{cases} 1+\beta_{i[t]} & \text{if} \quad \mathbf{a}_{i[t]} > 1 \\ -1+\beta_{i[t]} & \text{if} \quad \mathbf{a}_{i[t]} < -1 \\ (\delta+1)\mathbf{a}_{i[t]} - \delta\mathbf{a}^3_{i[t]} + \beta_{i[t]} & \text{otherwise,} \end{cases} \tag{7}$$

$$\beta_{i[t]} = \begin{cases} \gamma \tanh\left(\rho \mathbf{V}_{i[t]}\right) & \text{if} \quad \Delta E_{[t]} \le \varepsilon \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

$$\mathbf{V}_{i[t+1]} = \mathbf{V}_{i[t]} + \alpha \mathbf{x}_{i[t]}, \tag{9}$$

where γ, ρ, α are general output parameters, and $\mathbf{V}_{i[t]}$ is a vector to accumulate state values. ε is a threshold to judge whether the network is stable or not. $E_{[t]}$ is an energy function at time t and $\Delta E_{[t]}$ is difference between $E_{[t-1]}$ and $E_{[t]}$, and is expressed by following equation.

$$\Delta E_{[t]} = E_{[t-1]} - E_{[t]}, \tag{10}$$

$$E_{[t]} = -\frac{1}{2}\sum_{i=0}^{N}\sum_{j=0}^{N} w_{ij} x_{i[t]} x_{j[t]}. \tag{11}$$

This control neurons $\beta_{i[t]}$ give 0 if $\Delta E_{[t]}$ is larger than ε, or value of $\mathbf{V}_{i[t]}$ through the activation function if $\Delta E_{[t]}$ is smaller than ε in the network. When $\Delta E_{[t]} \le \varepsilon$, $\mathbf{V}_{i[t]}$ is initialized if $\mathbf{V}_{i[t]}$ is larger than threshold, otherwise $\mathbf{V}_{i[t]}$ is held as it is. For a working of $\beta_{i[t]}$, state of the network can be chaotically, when output is stable states.

2. Evaluation Simulation

We simulated to confirm effectiveness of proposed model. The stored pattern shown in Fig.3 is used, and the pattern with noise initial is given. Fig.6 shows the example of an output. From Fig.6, we see that the

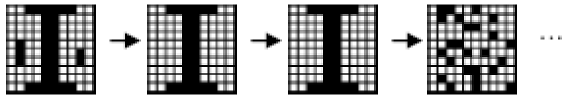
Fig.6. Example of an output

output changes dynamically after the network becomes steady state.

IV. COMPUTER SIMULATION

In this section, we carry out association simulation comparing proposed model to CNN proposed Aihara et al, confirm effectiveness of proposed model from points of storage capacity and associative success rate.

1. Conditions of Simulation

As stored patterns ($x_{[0]}$), we use the random pattern with the various correlation among stored patterns, the value is -1 or 1. Here, the number of neurons is $100(=10\times10)$, and these are generated by using the method used by Obayashi et al[2].

A. Learning stage

$\mathbf{W}_{[k]}(k=100)$ is learned by using equation (6)-(8). $\beta_{i[t]}$ is set to be 0 in the learning stage.

B. Association stage

In this paper, when all stored patterns were recall until 2000 steps (called one trial), it was assumed the associative success. The trial of 2000step was repeated 100 times independently, in every case, the initial value given to the network was generated at random. The identity frequency $h_{[t]}$ of the stored pattern and the output is introduced to judge the associative success, and it is defined as follows.

$$h_{[t]} = \frac{1}{N}\sum_{i=0}^{N} \mathbf{x}_{i[t]} * \mathbf{S}_i^{\mu}, \tag{12}$$

where $\mathbf{x}_{i[t]}$ is an output value of ith neuron (-1 or 1), $\mathbf{S}_i^{\mu}$ is ith element of μth stored pattern. Parameter are $\delta=0.1, \eta=0.03, \gamma=-2.0, \rho=0.9, \alpha=0.8, \varepsilon=0$. The number of learning trials was set to 100 times.

2. Storage Capacity Test

We examined storage capacity of the proposed model. As stored patterns, random pattern with correlation (=0.2) among stored patterns consisting of -1, 1 was given. The initial values of the network neurons were given at random. Fig.7 shows the associative success rates to stored patterns. As an object of comparison, we use the proposed model that uses our learning rule (6) explained in Section 3 and Chartier's learning rule (4) explained in Section 3, and CNN proposed by Aihara et al. From Fig.7, the proposed model that uses our learning rule (6) maintains a higher storage capacity than Aihara's CNN and the model with learning rule (4).

3. Association Test

To evaluate the association ability of the proposed model with Eq.(6), we changed the correlation of the stored pattern and simulated it. Fig.8 shows the result. From Fig.8, we can see the proposed model of the associative success rate is higher than Aihara's CNN in any case of 3, 4, and 5 stored patterns. Moreover, the associative success rate goes up in the proposed model when the correlation is high (for example, 0.8 and 0.9). It is thought that the reason is why the network state converges to near stored pattern easily, changing from a current stored pattern.

V. CONCLUSION

In this paper, we proposed 1) learning rule for associative matrix using gradient method, and 2) associative memory model of learning type that can escape from stable states. As a result, it had a high storage capacity and the high-speed recall ability, came to possible to escape from stable states, and it to recall a new pattern. In future works, we have to improve the proposed model (control neuron) and the application to the multi-valued image.

Appendix

We compute the changed portion $\Delta \mathbf{W}$ of learning rule from error function.

Given the linear output function as follows.

$$\mathbf{x}_{[1]} = \mathbf{W}\mathbf{x}_{[0]}$$

We define the error as follow.

$$\begin{aligned} e &= (\mathbf{x}_{[0]} - \mathbf{x}_{[1]})^T (\mathbf{x}_{[0]} - \mathbf{x}_{[1]}) \\ &= (\mathbf{x}_{[0]} - \mathbf{W}\mathbf{x}_{[0]})^T (\mathbf{x}_{[0]} - \mathbf{W}\mathbf{x}_{[0]}) \\ &= \mathbf{x}_{[0]}^T \mathbf{x}_{[0]} - \mathbf{x}_{[0]}^T \mathbf{W}\mathbf{x}_{[0]} - \mathbf{x}_{[0]}^T \mathbf{W}^T x_{[0]} - \mathbf{x}_{[0]}^T \mathbf{W}^T \mathbf{W}\mathbf{x}_{[0]} \\ &= \mathbf{x}_{[0]}^T \mathbf{x}_{[0]} - 2\mathbf{x}_{[0]}^T \mathbf{W}\mathbf{x}_{[0]} - \mathbf{x}_{[0]}^T \mathbf{W}^T \mathbf{W}\mathbf{x}_{[0]} \end{aligned}$$

We differentiate the error with respect to connection weight $\mathbf{W}$

$$\frac{de}{d\mathbf{W}} = -2\mathbf{x}_{[0]}\mathbf{x}_{[0]}^T + 2\mathbf{W}\mathbf{x}_{[0]}\mathbf{x}_{[0]}^T$$

We replace value -2 by a general learning parameter η.

$$\Delta \mathbf{W} = \eta (\mathbf{x}_{[0]}\mathbf{x}_{[0]}^T - \mathbf{x}_{[1]}\mathbf{x}_{[0]}^T)$$

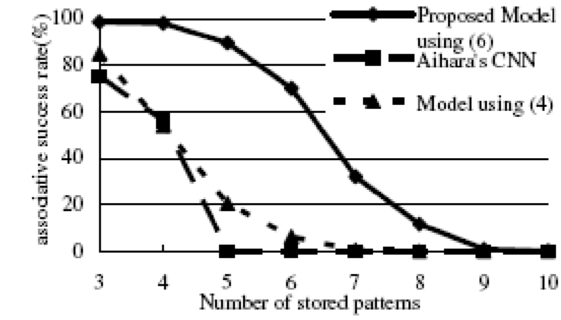

Fig.7. Storage capacities of each model

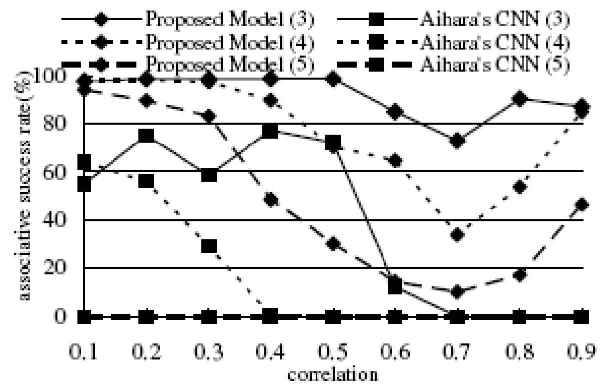

Fig.8 Success rate for various correlation in each stored patterns

Thus, we end up with the following learning rule.

REFERENCES

[1] Adachi M, Aihara K (1997), Associative Dynamics in a Chaotic Neural Network. Neural Networks, 10:83-98

[2] Obayashi M, Omiya R, Kuremoto T et.al (2006), Shapes of Non-monotonous Activation Functions in Chaotic Neural Network Associative Memory Model and Its Evaluation (in Japanese). IEEJ Transactions on Electronics, Information and Systems, 126-C (11):1401-1405

[3] S. Chartier, R. Proulx (2005), NDRAM: Nonlinear Dynamic Recurrent Associative Memory for Learning Bipolar and Nonbipolar Correlated Patterns. IEEE Trans. Neural Netw., 16(6):1393-1400

[4] S. Chartier, M. Boukadoum (2006), A Sequential Dynamic Heteroassociative Memory for Multistep Pattern Recognition and One-to-Many Association. IEEE Trans. Neural Netw., 17(1):59-68

[5] S. Chartier, M. Boukadoum (2006), A Bidirectional Heteroassociative Memory for Binary and Gray-Level Patterns. IEEE Trans. Neural Netw., 17(2):385-396