Vol. 42 No. SIG 5(TOM 4) IPSJ Transactions on Mathematical Modeling and Its Applications May 2001

Regular Paper

Mixed Pattern Segmentation by Using Chaotic Neural Networks

Yoshihisa Fukuhara† and Yoshiyasu Takefuji††

Many pattern segmentation systems have been developed by artificial neural networks. Those systems, including the perceptron and the associative memory are capable to map an input to one of the memorized patterns. However, in order to extract plural patterns from a mixed figure at once, we have to embed a specialized mechanism in their systems. The segmentation problem is still one of the difficult problems in image processing and speech recognition. This paper proposes a new pattern segmentation system using chaotic neural network. The proposed system can discriminate several patterns at once, even when they are overlapped each other. Our system is also reconfigurable, because it consists of two simple components: "back-propagation" and "chaotic neuron model" where the combination of two components plays a key role.

1. Introduction

The purpose of this study is to discriminate plural patterns from a mixed pattern by combining back-propagation and chaotic neural network. Many researchers have developed various kinds of pattern segmentation systems, for example, perceptron [1)] and associative memory [2)]. These methods accurately work, only if an input data does not include noises. However, in the case of practical use, information contains a lot of noises, where an input data is often incomplete and several patterns are overlapped each other. Especially, to discriminate the overlapped objects is an extremely difficult problem for image processing, because every pattern interferes each other. To resolve this problem, many researchers have developed a variety of systems.

Cho and Reggia proposed competitive and cooperative process [3)] derived from the error back-propagation for prediction of multiple disorders. Cohen and Grossberg developed a recognition method to discriminate mixed categories by using masking field [4)]. And Nigrin also developed SONNET [5)], which is a neural network system for this kind of tasks. Later, Basak et al. developed a category perception system, namely, X-tron [6)] which is a three-layered connectionist network model. And, Toyoda, et al. proposed associative neural networks with chaotic neurons [7)]. It can be used to discriminate mixed patterns by using chaotic neural networks [8)].

However, these conventional algorithms for recognizing a mixed pattern require a large amount of the prohibitive computational cost due to their complex strategy. Besides, these are sensitive to noise of input images. To resolve the fatal problem of their systems, we have developed a simple neural network by using the combination of "chaos neuron model" [9),10)] and the supervised learning algorithm "back-propagation" [11),12)]. Conventional back-propagation networks that use simple neuron models such as McCulloch-Pitts [13)] and sigmoid function output a fixed response from a static input. In fact, they show a good performance and stability. However, they only solve a static pattern recognition problem. By adding non-liner dynamics into back-propagation networks, the proposed system shows a dynamic behavior and solves more difficult pattern segmentation problems. In our approach, the main component of artificial neural networks is based on the chaotic neurons. Our system can extract each object from more than four mixed patterns simultaneously with a simple structure, and can be used for practical applications. The proposed system shows an advantage of the chaotic neural networks without instability of chaos. In the following section, we explain the mechanism of our system and the advantage of our system based on the simulation results.

2. The Proposed Method

The proposed system consists of two mechanisms, the chaos neuron model and the back-propagation algorithm. We describe the back-propagation algorithm and the chaos neuron

† Graduate School of Media and Governance, Keio University
†† Faculty of Environmental Information, Keio University

model. It is discussed how to obtain suitable parameters to control the chaos neuron. It is a significant process, although our system only works well with the suitable set of parameters of the chaos neuron.

2.1 Back-propagation

Back-propagation is a supervised learning algorithm proposed by Rumelhart in 1986 [11),12)]. It enables the multi-layer neural network (known as perceptron) to realize the segmentation of any nonlinear pattern. It gradually minimizes the average error based on a difference between an output and a teaching signal. The average error between the output and the teaching signal is given by Eq. (1). T is the teaching signal and O is the system output. The weight between jth neuron and ith neuron is updated by Eq. (2). Note that β and σ are constant values respectively.

$$E_p = \frac{1}{2}\sum_j (T_{pj} - O_{pj})^2 \tag{1}$$

$$\Delta_p W_{ji}(n+1) = \sigma\left(\frac{-\partial E_p}{\partial W_{ji}}\right) + \beta\Delta_p W_{ji}(n) \tag{2}$$

2.2 Chaotic Neural Networks

Aihara, et al. proposed a chaos neuron model [7),9),10)] which imitated a chaotic behavior caused by refractoriness of biological neuron. And they proposed a chaotic neural networks model based on the chaos neuron model. The dynamics of chaotic neural networks are given by Eq. (3), where $x_i(t+1)$ is the output of ith neuron at time $t+1$, M is the number of given data, N is the number of neurons, $A_j(t)$ is jth input value at time t, V_{ij} is the weight from $A_j(t)$ to ith neuron, W_{ij} is the weight from ith neuron to jth neuron, and α and θ_i are the refractory scaling parameter and the threshold. K_e, K_f, and K_r are the refractory decay parameters from external inputs, the feedback inputs, and the refractoriness, respectively. Equation (4) is the sigmoid function, where ϵ_1 and ϵ_2 define a slope of the sigmoid curve.

$$\begin{aligned} x_i(t+1) = f\Bigg(& \sum_{j=1}^{M} V_{ij}\sum_{d=0}^{t} K_e^d A_j(t-d) \\ & + \sum_{j=1}^{N} W_{ij}\sum_{d=0}^{t} K_f^d x_j(t-d) \\ & - \alpha\sum_{d=0}^{t} K_r^d x_i(t-d) - \theta_i \Bigg) \end{aligned} \tag{3}$$

$$f(u) = 1/(1+\exp((-u+\epsilon_1)/\epsilon_2)) \tag{4}$$

They also simplified Eq. (3) into Eq. (5). Although they used this network model for an associative memory, we applied this model to a multi-layered neural network.

$$\begin{aligned} x_i(t+1) &= f(\eta_i(t+1) + \zeta_i(t+1)) \\ \eta_i(t+1) &= K_f\eta_i(t) + \sum_{j=1}^{N} W_{ij}x_j(t) \\ \zeta_i(t+1) &= K_r\zeta_i(t) - \alpha x_i(t) + a_i \end{aligned} \tag{5}$$

where the term ζ is the input value and the term η is the mutual interactions, and a_i is the sum of threshold and temporally constant external inputs to ith neuron. Since the chaos neuron outputs an analog value, the neuron $x_i(t)$ in the output layer is converted to an alternative value based on Eq. (6).

$$x_i(t) = \begin{cases} 1 & (x_i(t) \geq 0.5) \\ 0 & (x_i(t) < 0.5) \end{cases} \tag{6}$$

2.3 Learning and Recognition

The proposed system is composed of a simple multi-layered neural network. In this paper, we use three layered neural networks as shown in **Fig. 1**. At the first stage, we use the back-propagation algorithm to teach several patterns to the network. In this stage, we use the sigmoid neuron described in Eq. (4). We do not use the chaos neuron model at the learning stage, because the proposed system does not need instability of the chaotic dynamism for learning. The chaos neurons are used when the system discriminate several patterns after learning.

In the next stage, we use the chaos neurons instead of the sigmoid neurons for pattern segmentation in order to recognize complex mixed patterns (Fig. 1). Note that the learning weights are kept at fixed values after learning.

Outputs of the system are recorded as sequential values during a certain period for one input. During the calculation period, we must not initialize the internal state in the chaos neu-

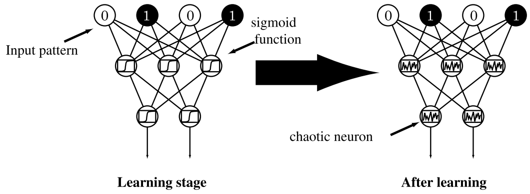


Fig. 1 The exchange of neurons.

rons. By keeping the internal state of the neurons, the chaos neurons enable the system to discriminate mixed patterns.

2.4 Optimization of the Chaos Neuron

Weights of neural networks are obtained by learning. However, the properties of a neuron, which control the chaotic dynamics, cannot be obtained by learning. Especially, the chaos neuron model is sensitive to the parameters of neuron. To make the system work reliable, we have to set correct values to the neuron models. To search for suitable parameters, we use Genetic Algorithms (GA) introduced by Holland[14].

K_f, K_r, α, and a_i as shown in Eq. (5) are the main control parameters of the chaos neuron. Accordingly, we defined these four parameters as genes. The following is an example optimization process to obtain suitable parameters.

(1) Initialization

We prepared 100 small multi-layered networks and each network has 400 inputs, 40 hidden neurons and 4 output neurons. The networks were trained for four patterns as shown in **Fig. 2** by using the backpropagation algorithm. In this process, we continued the learning until the average error $E < 0.002$. Note that $\beta = 0.90$, $\sigma = 0.25$ in Eq. (2) and the slope of the sigmoid curve $\epsilon_1 = 0.50$ and $\epsilon_2 = 1.00$. The initial values of K_f, K_r, α, and a_i were fixed at the uniform random values, and ϵ_1 and ϵ_2 in the sigmoid function of the chaos neuron model were fixed as 0 and 0.015 respectively.

(2) Selection Mechanism

We prepared for single input patterns and mixed input patterns, which are mixtures of any two patterns. These patterns were randomly chosen.

Each network was tested for these two kinds of inputs and we recorded a train of outputs at 100 times for each experiment. And next, we selected the individuals, which had a high percentage of correct answers with two experiments by using an elitist survival selection mechanism. The best 20% of all networks are used until the next generation and allowed reproduction.

(3) Operators

Two parents networks, which were randomly chosen produced 8 children. In this process, we used one-point crossover method. And the genes were randomly changed with 20% mutation rate. These processes were repeated for 20 generations and finally we obtained the parameters of chaotic neuron as shown in **Table 1**.

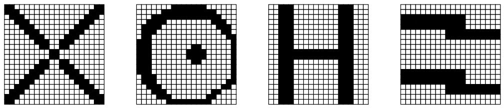


Fig. 2 Stored patterns.

Table 1 The parameters of chaotic neuron.

$K_f = 0.202511$	$a_i = 0.797882$
$K_r = 0.827443$	$\epsilon_1 = 0$
$\alpha = 2.136770$	$\epsilon_2 = 0.015000$

2.5 Handling of Outputs

To avoid interfering plural results each other, we should put a number of the neurons in the output layer and a number of memorized patterns in the same manner. And we should use one memorized pattern for one output neuron. For example, an output of neuron-1 must be 1 and other neurons must be 0, when the network learns Pattern 1. When the system discriminates many patterns, neurons in the output layer sometimes are activated simultaneously. This phenomenon causes a deterioration of rates of correct answers. Therefore, instead of rates of correct answers, we should consider firing rates of the neurons to obtain reliable answers. You should choose one from the counting methods and set up an appropriate threshold. In the next section, we show the firing rates of output neurons along with the percentages of correct answers.

3. Simulations and Results

To show the effectiveness of our system, we experimented three simulations on a computer. In this simulation, the network has 400 inputs, 40 neurons in the hidden-layer, and 4 neurons in the output-layer. And the system was trained to memorize four patterns as shown in Fig. 2. We continued the learning until the average error $E < 0.002$. Note that $\beta = 0.90$, $\sigma = 0.25$, $\epsilon_1 = 0.50$ and $\epsilon_2 = 1.00$. As a result, we recorded 100 sequential outputs for one static input. In this paper, we defined a noise as a bit inversion.

3.1 Simulation-1: Input a Single Pattern

When we gave Pattern 1 to the network, the system shows a correct answer perfectly as shown in **Table 2**. **Figure 3** shows the state of neurons in the output layer from the time scale

Table 2 The results of simulation-1.

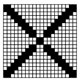
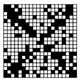
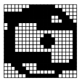
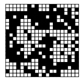
Input	Percentages of correct answeres	Firing rates
 PATTERN 1	Pattern 1 = 100% Pattern 2 = 0% Pattern 3 = 0% Pattern 4 = 0% Unknown = 0%	Output 1 = 100% Output 2 = 0% Output 3 = 0% Output 4 = 0%
 PATTERN 1 with 20% noise	Pattern 1 = 100% Pattern 2 = 0% Pattern 3 = 0% Pattern 4 = 0% Unknown = 0%	Output 1 = 100% Output 2 = 0% Output 3 = 0% Output 4 = 0%

Table 3 The results of simulation-2.

Input	Percentages of correct answeres	Firing rates
 PATTERN 2+4	Pattern 1 = 0% Pattern 2 = 50% Pattern 3 = 0% Pattern 4 = 49% Unknown = 1%	Output 1 = 0% Output 2 = 51% Output 3 = 0% Output 4 = 50%
 PATTERN 2+4 with 20% noise	Pattern 1 = 0% Pattern 2 = 50% Pattern 3 = 0% Pattern 4 = 49% Unknown = 1%	Output 1 = 0% Output 2 = 51% Output 3 = 0% Output 4 = 50%

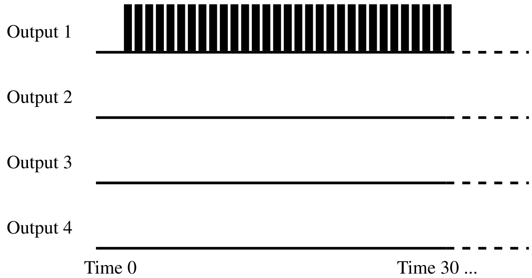


Fig. 3 State of output neurons of simulation-1. (Input: Pattern 1)

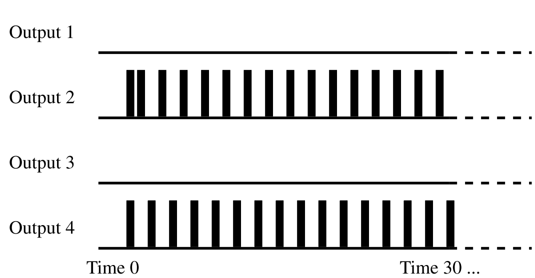


Fig. 4 State of output neurons of simulation-2. (Input: Pattern 2 + 4)

0 to 30. Although we gave an input with 20% noise, we obtained the perfect answer shown in Table 2. We discuss this issue more in the next section.

3.2 Simulation-2: Segmentation of Two Patterns

When we gave a mixed pattern of Pattern 2 and Pattern 4, the proposed system showed two answers mutually as shown in **Table 3** and **Fig. 4**. Moreover, the system showed almost perfect answer in this simulation, although the input included a large amount of noises.

3.3 Simulation-3: Segmentation of Three Patterns

When we gave a mixed pattern of Pattern 2, Pattern 3, Pattern 4, the proposed system did not show a good performance seemingly. However, you can observe that the proposed system recognize each of three patterns from four memorized patterns, when we observe state of the neurons in the output layer as shown in **Fig. 5**. In Fig. 5, output neuron-3 and output neuron-4 was activated simultaneously. The proposed system shows an excellent answer from the view of firing rates as shown in the upper part of **Table 4**. Finally, we gave a mixed pattern of the three patterns with noises, the proposed system showed accurate solutions from the difficult input value as shown in the lower part of Table 4 and **Fig. 6**.

4. Performance Analysis

In this section, we discuss advantages of our system from a variety of viewpoints.

4.1 Noise Tolerance

We examined the noise tolerance of our system. **Figure 7** shows the noise tolerance for a single input pattern. In this simulation, we used the same network and the same computational condition as shown in Section 3. In this simulation, the chaos neuron c, which shows a correct answer, satisfies $R_c = MAX\{R_i\ (i = 1 \ldots N)\}$. Note that R_i is the firing rate of ith neuron. As shown in Fig. 7, the noise tolerance of the proposed system is superior to the conventional sigmoid system. The proposed system succeeds in distinguishing a mixed pattern as keeping the same ability of the sigmoid system in terms of the noise tolerance. **Figure 8** shows state of the output neurons, when the network obtains a mixture of two patterns. We conclude that the proposed system can extract mixed figure accurately if an amount of noise is up to 40%.

4.2 Calculation Speed

We computed the calculation speed of the proposed system. **Table 5** shows the performance of the computer used in our simulation. **Table 6** shows learning speeds and calcula-

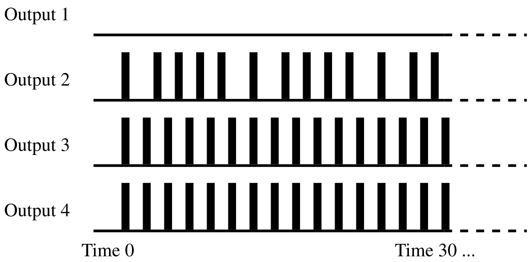


Fig. 5 State of output neurons of simulation-3. (Input: Pattern 2 + 3 + 4)

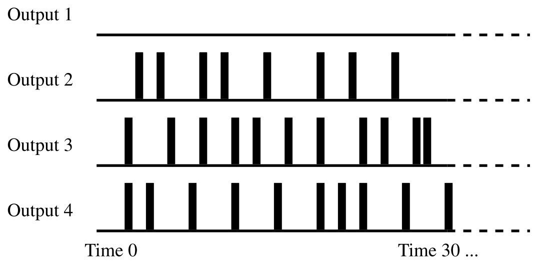
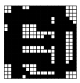
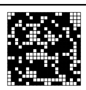


Fig. 6 State of output neurons of simulation-3. (Input: Pattern 2 + 3 + 4 with 20% noise)

Table 4 The results of simulation-3

Input	Percentages of correct answeres	Firing rates
 PATTERN 2+3+4	Pattern 1 = 0% Pattern 2 = 32% Pattern 3 = 0% Pattern 4 = 0% Unknown = 68%	Output 1 = 0% Output 2 = 41% Output 3 = 50% Output 4 = 50%
 PATTERN 2+3+4 with 20% noise	Pattern 1 = 0% Pattern 2 = 20% Pattern 3 = 17% Pattern 4 = 17% Unknown = 46%	Output 1 = 0% Output 2 = 28% Output 3 = 34% Output 4 = 30%

tion speeds for segmentation of two different kinds of networks. In this simulation, we used 30 sequential outputs as a sufficient condition for segmenting any patterns. And we continued the learning process until the average error $E < 0.002$. The calculation of the proposed system is extremely fast as shown in Table 6. Our system discriminates input patterns within 30 ms for 400 input values. This calculation cost satisfies a performance for the real time information processing.

4.3 Extension Ability

(1) Structure of the network

We examined various sizes of networks and changed the number of neurons in a hidden layer. For example, we examined the same simulation as the simulation-3 changing the number of the hidden neurons between 5 and 70. And we obtained almost same results from all examinations.

(2) The number of stored patterns

We examined changing the number of stored patterns. As shown in **Table 7** and **Fig. 9**, the proposed system extracted four patterns at once, when the network learns five patterns as shown in **Fig. 10**. From this simulation,

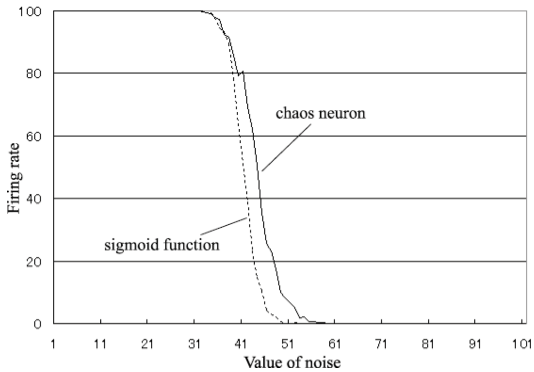


Fig. 7 Noise tolerance with a single input.

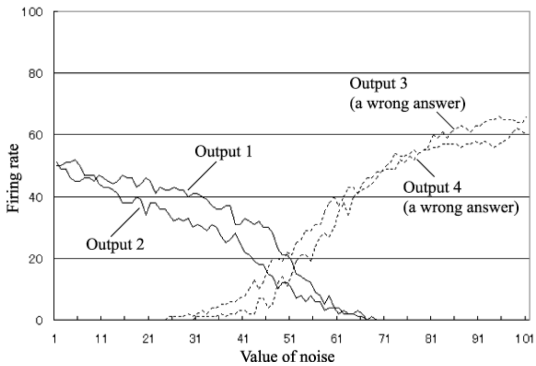


Fig. 8 Noise tolerance with a mixed input.

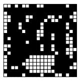
Table 5 The environment of PC performance.

CPU	Pentium II 400 Mhz
Memory	128 MB
Operating System	Windows NT
Development Language	Borland Delphi 4.0

Table 6 Calculation speeds.

Network Configuration	Learning Speed	Segmentation Speed (30 times)
Input: 100 neurons Hidden: 10 neurons Output: 3 neurons	390 msec	3 msec or below
Input: 400 neurons Hidden: 20 neurons Output: 4 neurons	2954 msec	27 msec

Table 7 Discrimination of four patterns.

Input	Percentages of correct answeres	Firing rates
 PATTERN 1+3+4+5 with 10% noise	Pattern 1 = 0% Pattern 2 = 0% Pattern 3 = 16% Pattern 4 = 0% Pattern 5 = 17% Unknown = 67%	Output 1 = 33% Output 2 = 0% Output 3 = 33% Output 4 = 33% Output 5 = 50%

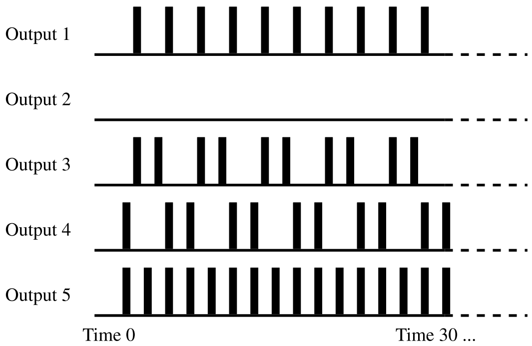


Fig. 9 State of output neurons. (Input: Pattern 1 + 3 + 4 + 5 with 10% noise)

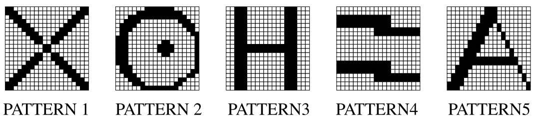


Fig. 10 Stored patterns.

we can understand if a correlation between stored patterns is low enough, the proposed system can discriminate patterns from a mixture of more than four patterns at once.

5. The Network Analysis

In this section, we discuss the property of chaotic neural network with suitable parameters and consider why the proposed system can discriminate mixed figures.

To analyze the behavior of the chaotic neural network with suitable parameters used in the proposed system, we use a simple neural network, which consists of two chaos neurons as shown in **Fig. 11**. X_1 and X_2 are the chaotic neurons, I is an input value and W is a weight between X_1 and X_2. **Figure 12** shows an output value from the neuron X_1, when the input value I was changed. As shown in Fig. 12, in the most of cases, the neuron X_1 outputs only 0 or 1, although the network shows the bifurcation in places. **Figure 13** shows a firing rate of neuron X_1 and **Fig. 14** shows activations of neuron X_1. As shown in Fig. 14, the state converges

Fig. 11 A chaotic neural network composed of two neurons.

to a limit cycle at flat places in Fig. 13. From these results, we can understand the chaos neuron, which is used in the proposed system shows periodic activation in most cases. Next, we observed the behaviors of neuron X_2. **Figure 15** shows a firing rate of neuron X_2, when I and W were changed. The firing rate shows 0.5 in large part. In addition, the graph shows flat surfaces in the most of the rest. We think X_2 is activated periodically under these conditions.

From these results, by combining two chaotic components, the chaotic dynamics of the neurons is attenuated and the network shows periodic activation. We think this periodic activation of the neurons influence the pattern segmentation of mixed figures.

6. Discussion and Conclusions

In this paper, we propose a new pattern segmentation system, which is capable of discriminating several patterns simultaneously from a mixed pattern.

SONNET [5] and X-tron [6] are also able to discriminate mixed patterns by using self-organization. Compared with these conventional methods, the construction of the proposed method is simple. Since our system consists of the "back-propagation" method and the three-layered neural network, we can handle and modify it without hard tasks. Moreover, the proposed system can discriminate more than four patterns at once with a high tolerance for noises. Cho and Reggia proposed a learning scheme for this kind of purpose [3]. It consists of a simple architecture, which is similar to our system. However, they did not describe to the influence of noise. Thus, our system shows a new possibility of back-propagation by using chaotic neural network.

Generally, neural network systems using chaotic dynamism show an unstable behavior. As the system, which uses the chaotic neural networks, Toyoda, et al. proposed associative neural networks using chaotic neurons [7]. Their system can associate plural patterns chaotically, and if a mixed pattern is presented, the system shows related categories with high fre-

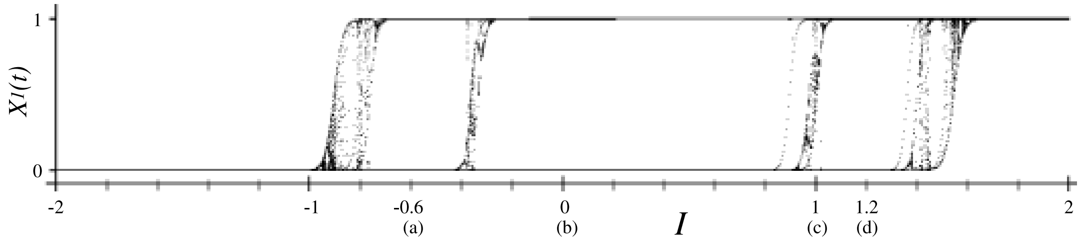


Fig. 12 Output of chaos neuron X_1.

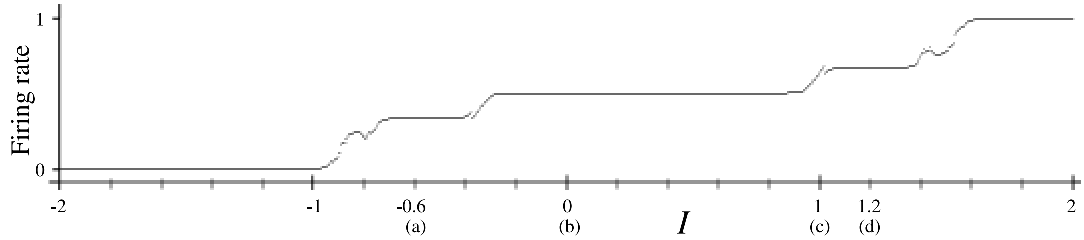


Fig. 13 Firing rate of chaos neuron X_1.

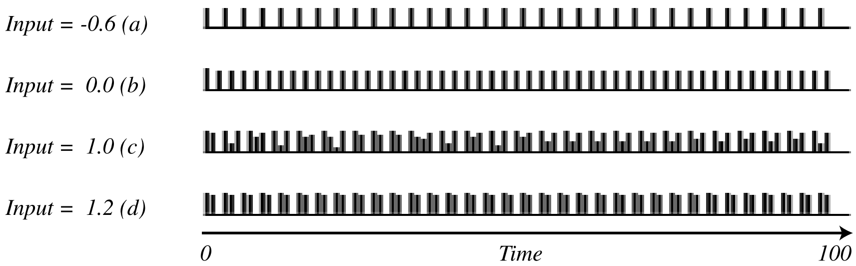


Fig. 14 Firing behavior of chaos neuron X_1.

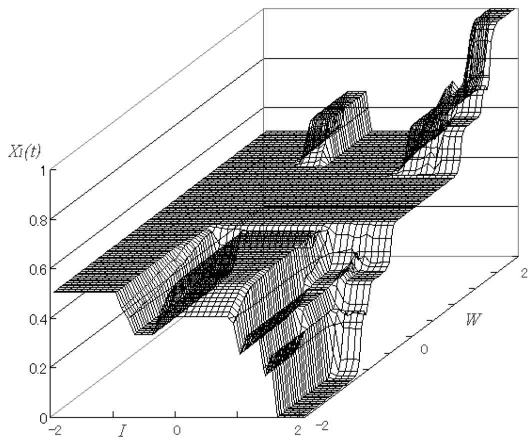


Fig. 15 Firing rate of chaos neuron X_2.

quency[8]. However, it still remains a hard problem to control the non-linear dynamics to discriminate more than two patterns.

Since the proposed system shows a reliable performance and stability by tuning the values of parameters to control the chaos neurons, we think our system can be used for complex information processing including video observation system, speech recognition, and text recognition.

7. Future Works

The proposed system shows extremely a good result, although an input to perceptional recognition often contains a kind of noises, such as the rotation and orientation of input image. As a future work, we have to normalize the input value, and improve the model so that it can cope with rotation and orientation of input image.

Acknowledgments We would like to thank Dr. Souichi Oka for his support and helpful advice.

References

1) Minsky, M. and Papert, S.: *Perceptrons*, MIT Press (1969).
2) Hopfield, J.: Neural Networks and Physical Systems with Emergent Collective Computa-

tional Abilities, *Proc. National Academy of Science*, No.79, pp.2445–2558 (1982).

3) Cho, S. and Reggia, J.: Learning competition and cooperation, *Neural Computa.*, Vol.5, pp.242–259 (1993).
4) Cohen, M. and Grossberg, S.: Masking fields: A massively parallel neural architecture for learning, recognizing, and predicting multiple groupings of data, *Appl. Opt.*, Vol.26, pp.1866–1891 (1987).
5) Nigrin, A.: SONNET: A self-organizing neural network that classifies multiple patterns simultaneously, *Conf. Neural Networks*, Washington. DC, pp.525–528 (1990).
6) Basak, J. and Pal, S.K.: X-tron: An Incremental Connectionist Model for Category Perception, *IEEE Trans Neural Networks*, Vol.6, No.5, pp.1091–1108 (1995).
7) Toyoda, M., Aihara, K., Shimizu, K., Adachi, M. and Kotani, M.: Chaotic neural networks, *Proc. SICE '89*, pp.1323–1325 (1989).
8) Adachi, M., Aihara, K. and Cichocki, A.: Separation of mixed patterns by chaotic neural networks, *1993 International Symposium on Nonlinear Theory and Its Applications—NOLTA '96* (*Kochi*), *NTA Research Society of IEICE*, pp.93–96 (1996).
9) Aihara, K., Takabe, T. and Toyoda, M.: Chaotic Neural Networks, *Phys. Lett. A*, Vol.144, pp.333–340 (1990).
10) Aihara, K.: Chaotic Neural Networks, Kawakami, H. (Ed.), *Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems*, pp.143–161, World Scientific (1990).
11) Rumelhart, D.E., Hinton, G.E. and Williams, R.J.: Learning representaions by back-propagating errors, *Nature*, Vol.323, No.9, pp.533–536 (1986).
12) Rumelhart, D.E., Hinton, G.E. and Williams, R.J.: Parallel distributed processing: Explorations in the microstructure of cognition, Rumelhart, D.E., McCelland, J.L. and PDP Research Group (Eds.), *Learning internal representations by error propagation*, Vol.1, Chap.8, pp.318–362, MIT Press (1986).
13) McCulloch, W. and Pitts, W.: A logical calculus of ideas immanent in nervous activity, *Bulletin of Mathmatical Biophysics*, Vol.5, pp.115–133 (1943).
14) Holland, J.: *Adaptation in Natural and Artificial Systems*, University of Michigan Press (1975).

(Received August 25, 2000)
(Revised October 16, 2000 (1st))
(Revised November 23, 2000 (2nd))
(Accepted November 28, 2000)

Yoshihisa Fukuhara was born in 1975. He received his master's degree from Graduate School of Media and Governance, Keio University in 1999 and he has been a doctoral student of Keio University since 2000. His research interests include the application of neural networks and non-liner dynamics. He is a student member of IPSJ. Publication: Takashi Iba, Yoshihisa Fukuhara, Fukuzatsukei Nyumon [Introductory Complex Systems: An Adventure to the Frontier of Knowledge], NTT Publishers, 1998 (in Japanese).

Yoshiyasu Takefuji is a professor of Keio University from 1992, was on the faculty of Case Western Reserve Univ. since 1988, Univ. of South Carolina (1985–1988), Univ. of South Florida (1983–1985). Research: neural computing and hyperspectral computing. Awards: NSF-RIA in 1989, distinguished service from IEEE Trans. on NN in 1992, best paper of IPSJ in 1980 and that of IFAC at AIRTC '98, TEPCO and KAST in 1993, Takayanagi award in 1995, KDD award in 1997, NTT tele-education courseware award in 1999. Government advisor: NCC of the Philippines, VITTI of Vietnam, Jordan IT, Multimedia Univ. in Malaysia. Published: 16 books and more than 160 papers.