Brain Inspired Cognitive Systems
August 29 - September 1, 2004
University of Stirling, Scotland, UK

ON REFRACTORINESS OF CHAOTIC NEURONS IN INCREMENTAL LEARNING

Toshinori Deguchi, Gifu National College of Technology, Gifu, 501–0495, Japan

Naohiro Ishii, Aichi Institute of Technology, Aichi, 470–0392, Japan

ABSTRACT

This paper develops the incremental learning by using chaotic neurons, which is called "on-demand learning" at its developing time. The incremental learning unites the learning process and the recall process in the associative memories. This learning method uses the features of the chaotic neurons which were first developed by Prof. Aihara. The features include the spatio-temporal sum of the inputs and the refractoriness in the chaotic neurons. Because of the temporal sum of the inputs, the network learns from inputs with noises. But, it is not obvious that all the features are needed to the incremental learning. In this paper, the computer simulations investigate how the refractoriness takes an important part in the incremental learning. The results of the simulations, show that the refractoriness is an essential factor, but that strong refractoriness causes failures to learn patterns.

1 INTRODUCTION

It is well known that associative memory can be performed in the neural networks by using the correlative learning. In the correlative learning, the learning process and the recall process are usually separated, because a neural network used in the correlative learning usually learns patterns at first, then it recalls one of the patterns when an input is given.

In the incremental learning, which is called on-demand learning at its developing time, the learning process and the recall process are united[1].

In this learning, the network keeps receiving the inputs. If the inputs are known to the network, it recalls them. Otherwise, each neuron in the network learns them gradually. The basic idea of the incremental learning is from the automatic learning[2]. As same as in the incremental learning, in the automatic learning, the neurons decide whether an input is known or not by themselves and learn. But, the automatic learning have 4 threshold values for the learning which are difficult to determine.

Each neuron decides when it learns out of its internal informations which are the inputs to the neuron and the internal states of the neuron. The neurons used in this learning are the chaotic neurons, and their network is called the chaotic neural network which was developed by Prof. Aihara[3].

The chaotic neuron has temporal sum of inputs, which enables the network to learn from noisy inputs.

The chaotic neuron also has refractoriness. The refractoriness is used in the learning, but it is not obvious whether the refractoriness is essential to the incremental learning.

In this paper, the computer simulations investigate how the refractoriness plays an important role in the incremental learning.

2 INCREMENTAL LEARNING

The incremental learning uses Hopfield's type network. Each neuron in the network receives the signals from the other neurons in the network and the signal from the external inputs through the connection weights. This type of network has been used in associative memory, except that the external inputs of this network are kept sending the input patterns continuously.

The incremental learning was developed by using the chaotic neurons. The chaotic neuron and the chaotic neural network were developed by Prof. Aihara.[3]. In the chaotic neural network, the spatio-temporal sum is introduced and the refractoriness is considered as a negative feedback. The chaotic neurons in the chaotic neural network depend on the dynamics as follows:

$$x_i(t+1) = f[\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)] \quad (1)$$

$$\begin{aligned} \xi_i(t+1) &= k_s \xi_i(t) + \upsilon A_i(t) \\ \eta_i(t+1) &= k_m \eta_i(t) + \sum_{j=1}^{49} w_{ij} x_j(t) \\ \zeta_i(t+1) &= k_r \zeta_i(t) - \alpha x_i(t) \end{aligned} \quad (2)$$

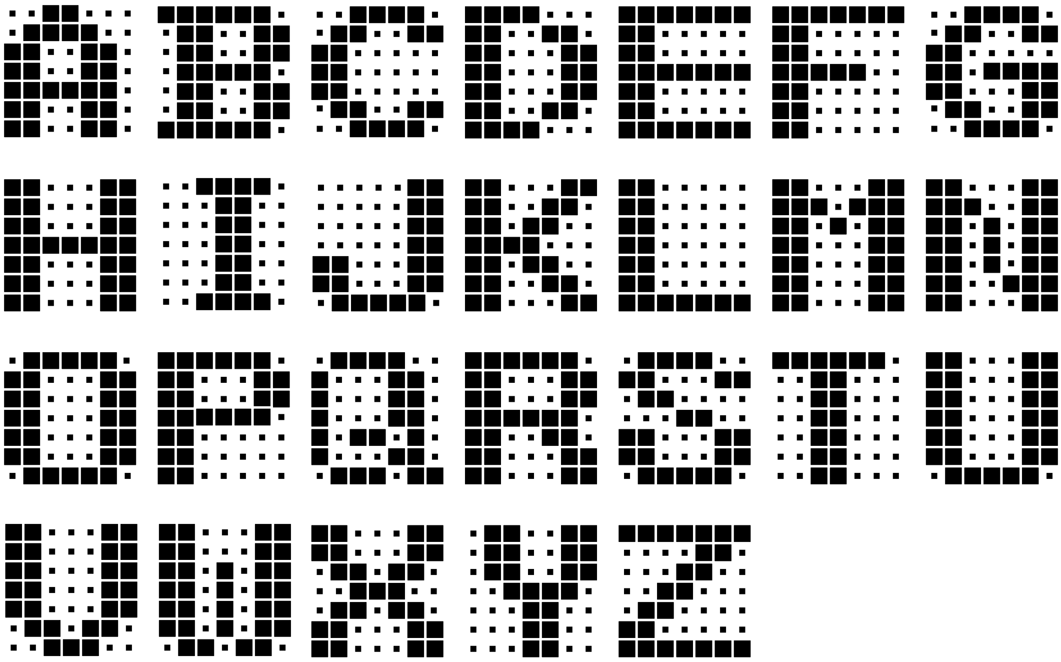


Figure 1: Patterns to be learnt in the network

Table 1: Parameters

$\upsilon = 2.0$	$k_s = 0.95$	$k_m = 0.1$	$k_r = 0.95$
$\alpha = 2.0$	$\theta_i = 0$	$\varepsilon = 0.015$	

where $x_i(t+1)$ is the output of i-th neuron at time $t+1$, f is the output sigmoid function described below in Equation (3), $A_i(t)$ is the input to i-th neuron at time t, k_s, k_m, k_r are time decay constants, υ is the weight for external inputs, w_{ij} is the connection weight from neuron j to neuron i, and α is the parameter that specifies the relation between the neuron output and the refractoriness. In Equation (2), ξ, η, and ζ show external input, mutual interaction, and refractoriness respectively.

$$f(x) = \frac{2}{1+\exp(\frac{-x}{\varepsilon})} - 1 \tag{3}$$

In this paper, the network is composed of 49 chaotic neurons. The parameters in the chaotic neurons are assigned to the values in Table 1. The 26 input patterns are shown in Figure 1. In each pattern, 49 inputs are arranged by 7×7 and a large black square represents 1 and a small one does -1.

The network has each pattern inputted during 50 steps, before moving to the next one. After all the patterns are shown, the first pattern comes repeatedly. A set is defined as a period through 26 patterns from the first pattern to the last pattern. It means that one set is 50×26 steps in these simulations.

The network does not make the neurons change their connection weights simultaneously. Each neuron decides whether it changes the connection weights or not. To decide it, the neuron checks the learning condition shown in Equation (4).

$$\xi_i(t) \times (\eta_i(t) + \zeta_i(t)) < 0 \tag{4}$$

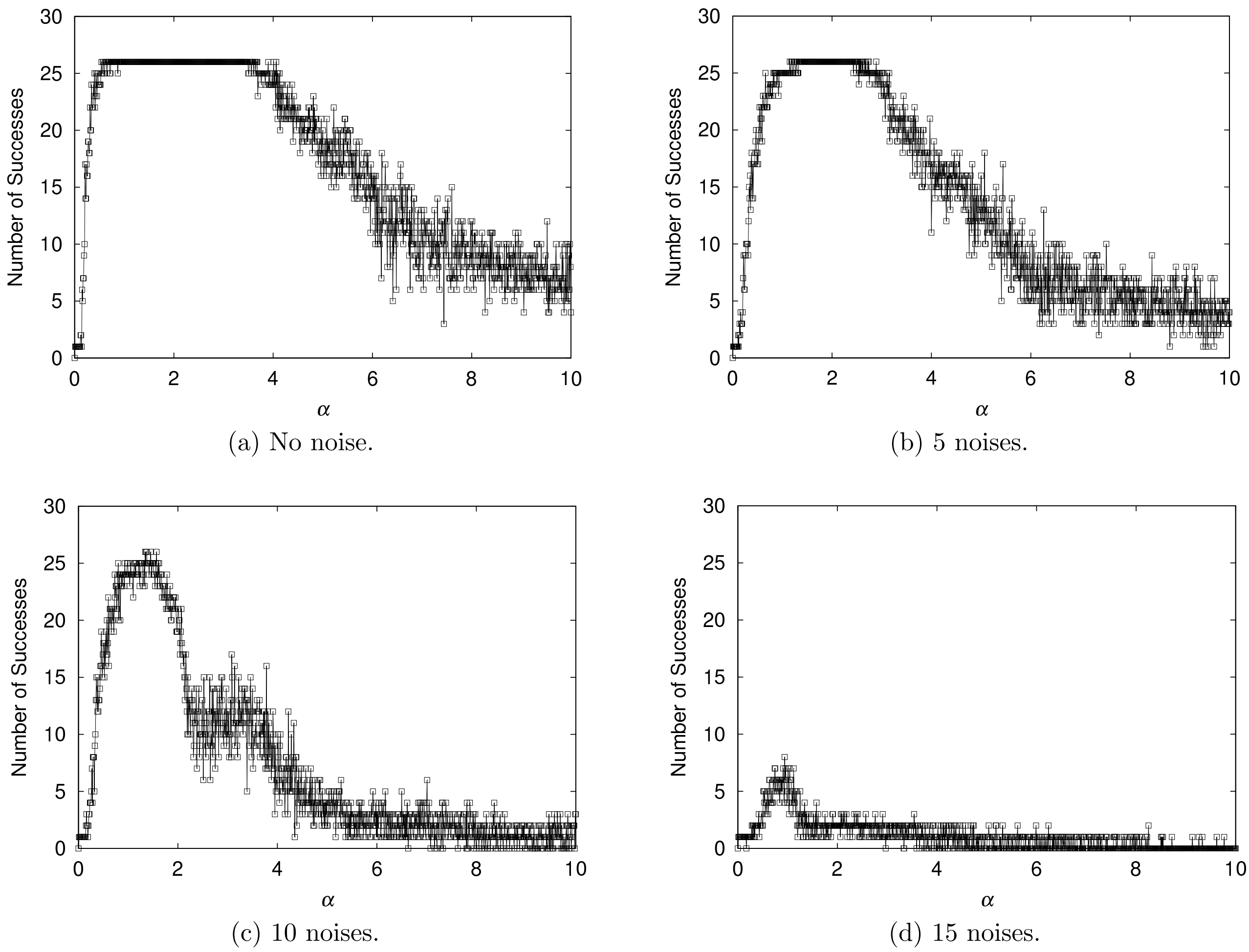
When the network knows an input pattern, the external input and the mutual interaction have same signs in every neurons. When the network doesn't know an input pattern, the external input—$\xi_i(t)$—and the mutual interaction—$\eta_i(t)$—have different signs in some neurons. In this learning method, each neuron changes its connection weights, if the mutual interaction has different signs from the sign of the external input. To make the patterns memorized firmly, if the mutual interaction is less than the refractoriness—$\zeta_i(t)$— in the absolute value, each neuron also changes its connection weights.

When the condition is satisfied, the neuron changes its connection weights as follows:

$$w_{ij} = \begin{cases} w_{ij} + \Delta w, & \xi_i(t) \times x_j(t) > 0 \\ w_{ij} - \Delta w, & \xi_i(t) \times x_j(t) \leq 0 \end{cases} \tag{5}$$

This is based on the simple rule to increase the weight when the external input of the neuron and the output of the other neuron have the same sign, or to decrease when they have different signs.

The repetition of the conditions and the connection-changes makes the network have the input pattern memorized. In this learning, the initial values of the connection weights can be 0, because some outputs of neurons are changed by their external inputs and this makes the condition establish in some neurons. In this paper, all of the initial values of the connection weights are set to 0.



(a) No noise. (b) 5 noises. (c) 10 noises. (d) 15 noises.

Figure 2: Refractory parameter α vs number of successes

3 REFRACTORINESS OF CHAOTIC NEURONS

The chaotic neurons have some features which the usual neurons don't have. In this paper, the features of refractoriness are focused.

It is not obvious whether the refractoriness is needed in the incremental learning or not. Each neuron changes its connection weights along with two inputs. One is the external input from the outside of the network. The other is the input with the mutual interaction. The basic concept of changing weights of this learning is to match these inputs in the same sign—positive or negative. Therefor, these inputs should be important. At developing time, the refractoriness was intended to make the patterns memorized firmly.

3.1 LEARNT PATTERNS BY REFRACTORY PARAMETERS

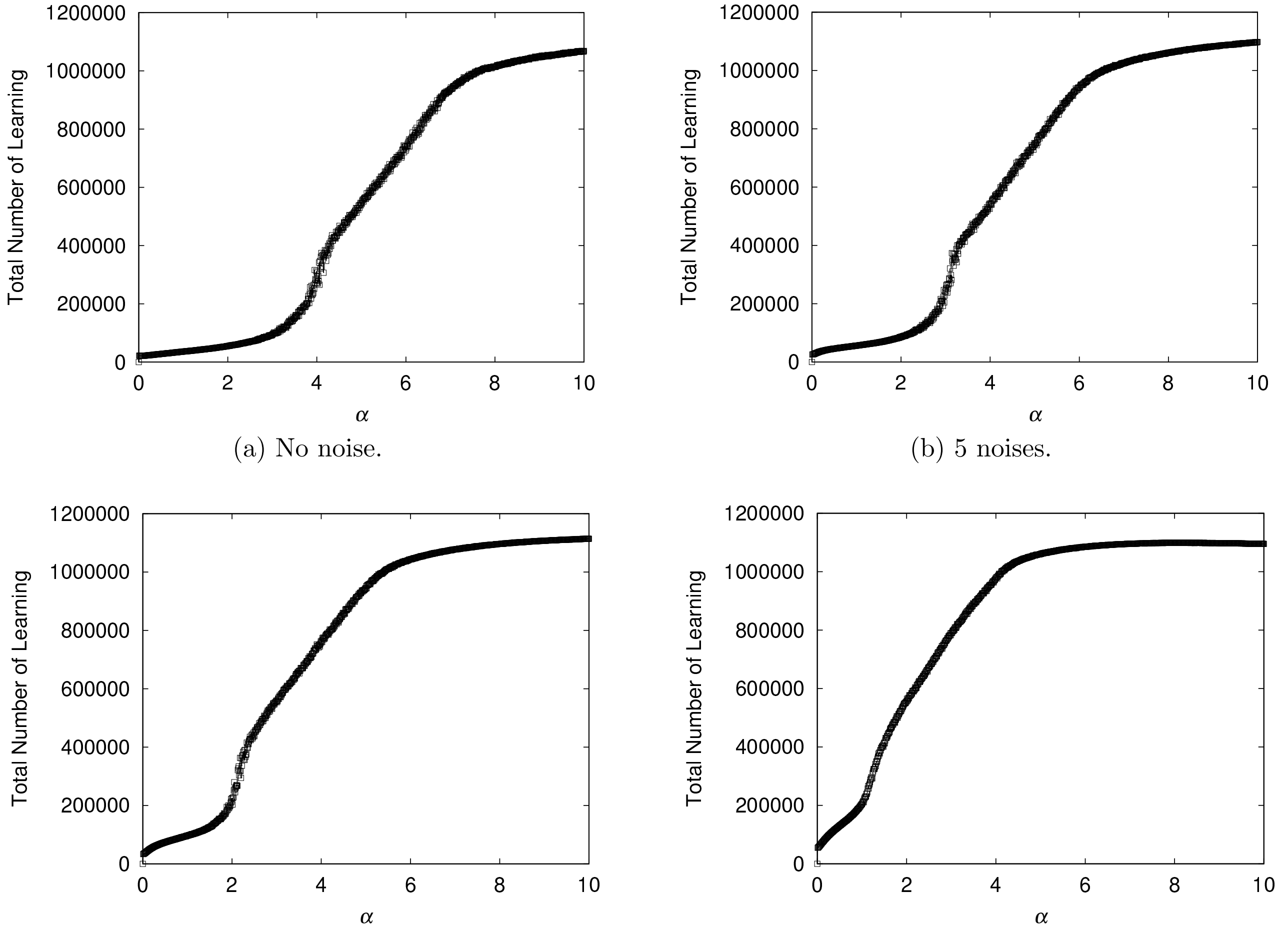
Computer simulations were carried out to make clear the role of the refractoriness by changing the parameter. When $\alpha = 0$, the refractoriness does not exist. In the former work, α was settled to 2.0[1]. In this paper, α was swept from 0.0 to 10.0 to investigate the effects of the refractoriness.

In the incremental learning, the network can learn from noisy inputs, because the external input is summed in time and the network mainly receives the effects of correct inputs. For the noisy conditions, 5, 10, or 15 elements are selected at random at each step and turned 1 to -1 or -1 to 1.

The result after 20 sets is shown in Figure 2. In each graph, the horizontal axis shows the value of α and the vertical axis shows the number of the patterns learnt successfully. It has to be notified that the associative memory with the correlative learning could not learn any of the patterns when all 52 patterns are inputted without noises.

Figure 2 shows that no pattern was stored in the network when there is no refractoriness ($\alpha = 0$). From $\alpha = 0$ to $\alpha = 0.9$, the number of success is increasing steeply. Within 5 noises, for the range near $\alpha = 2.0$, all 26 patterns are successfully memorized. This means that the refractoriness is essential for this incremental learning.

Figure 2(b) to (d) show that all the numbers of success in this range are growing down as the number of noises



(a) No noise.

(b) 5 noises.

(c) 10 noises.

(d) 15 noises.

Figure 3: The α vs total number of learning

increases. But it is remarkable that in the case of 10 noises in Figure 2(c), the network could learn all 26 patterns at some values of α. 10 noises are more than 20% of 49 neurons.

In the range over $\alpha = 3.5$, the value of α becomes bigger, the number of success becomes more decreasing in Figure 2(a). The strong refractoriness causes failures in the learning of patterns. Following simulation results show how the learning goes to failure, along with α.

3.2 REFRACTORY PARAMETER VS TOTAL NUMBER OF LEARNING

To investigate the failure in the high value of α, the computer simulations count how many times the neurons satisfy the learning condition. As same as the former simulation, α was swept from 0.0 to 10.0 to count the total number of learning.

The result after 20 sets is shown in Figure 3. In each graph, the horizontal axis shows the value of α and the vertical axis shows the total number of learning.

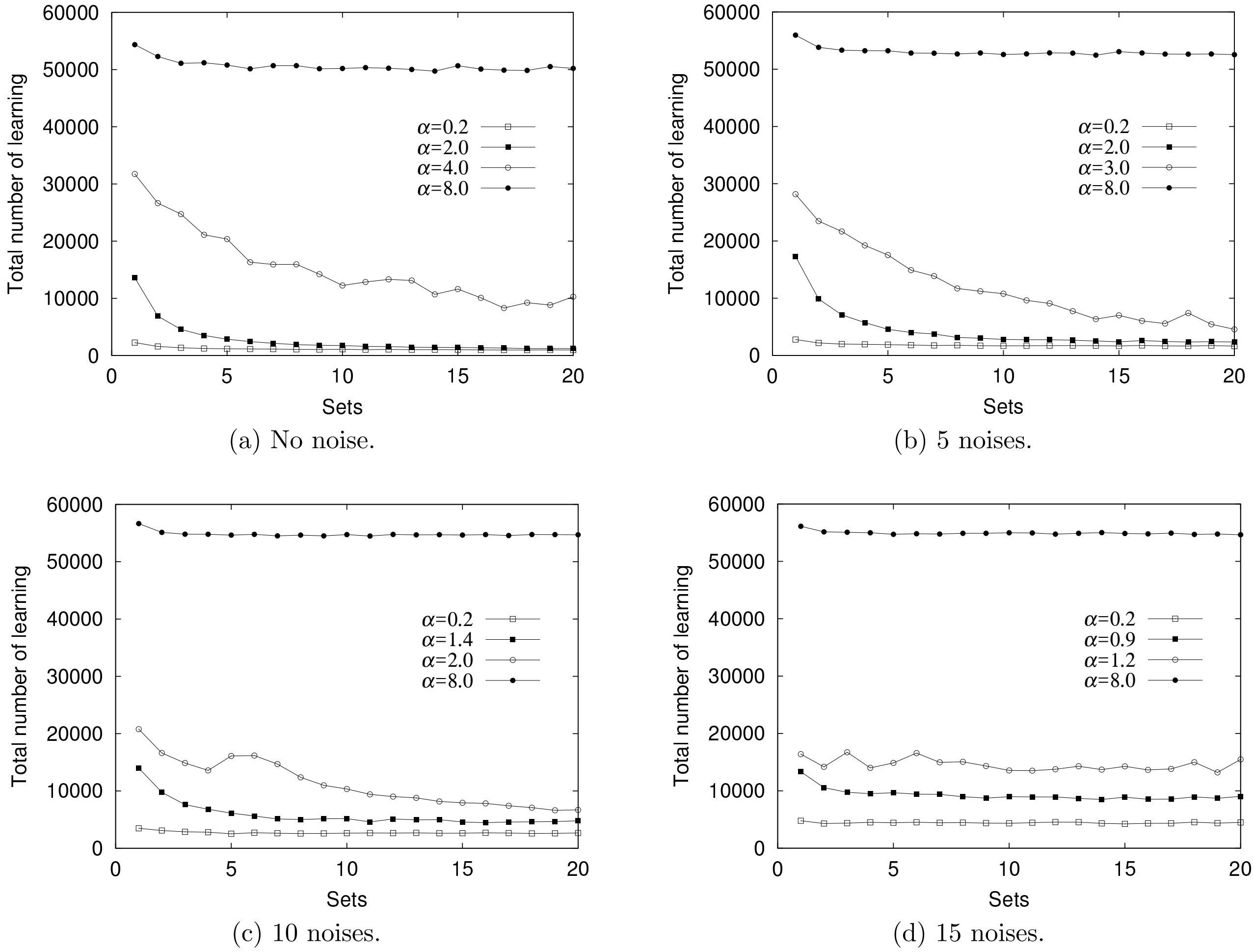
According to the value of α growing, the total number of learning increases.

In Figure 3(a), in the range under $\alpha = 3.5$, the total number of learning increases gradually. Comparing to Figure 2, the neurons learn more, as the refractoriness grows. From this result, it is a reasonable consideration that the number of success is getting bigger from $\alpha = 0$ to $\alpha = 0.9$ because the total number of learning increases, and that it saturates from $\alpha = 0.9$ to $\alpha = 3.5$. The upper limit is 1274000—49 neurons $\times$ 50 step par each pattern $\times$ 26 patterns $\times$ 20 sets.

While the total number of learning increases steeply over $\alpha = 3.5$, the number of success is getting smaller.

Figure 3(b) to (d) show that all the total number of learning increases more rapidly, as the number of noises increases. The noises probably make the learning condition establish even when the learning is not needed.

For more considerations, the next computer simulations investigate how the total number of learning in one set varies along with sets.



(a) No noise. (b) 5 noises. (c) 10 noises. (d) 15 noises.

Figure 4: Total number of learning in each set without noises

3.3 TOTAL NUMBER OF LEARNING IN EACH SET

In these simulations, to investigate the failure in the high value of α, the total numbers of learning in one set are counted in each set. A set has been defined as a period of 50×26 steps in this paper. In each noise condition, α is set to four different values. The first is where α is low, the second is where the number of success becomes high in Figure 2, the third is where the number of success becomes steep in Figure 3, and the fourth is where α is too big.

The result is shown in Figure 4. The horizontal axis shows the sets. The vertical axis shows the total number of learning in each set.

In Figure 4(a), the total numbers of learning are getting smaller, in the successful value of $\alpha = 2.0$, as the time is passed by in the network. Because the network has memorized no pattern at first, the neurons learn frequently. When the network gets memorized the patterns, the neurons don't learn frequently any more.

Comparing Figure 4(a) to (d), the curves become flatter as the noises increase. This means that the network has not learnt the patterns. Therefore, the noises with high occurring rate must prevent the network to learn correctly. At high noise occurring rate, the noises could change the signs of the external inputs and make the learning condition satisfied, then the neurons change their connection weights to the wrong way.

In the low value of $\alpha = 0.2$, it is obvious that the total numbers of learning are too small to learn.

In the high values of $\alpha = 4.0$ or 8.0, the total numbers of learning are high, especially $\alpha = 8.0$. In the case of $\alpha = 8.0$, the numbers are almost same. There are two hypothetical reasons that can be thought.

One is that the learning is so strong that the network forgets the former patterns, because most patterns that the network didn't learn in the high α are in the head part of the patterns as shown in Figure 5. In Figure 5, the horizontal axis shows the value of α and the vertical axis represents the patterns to be learnt. The top of these patterns is "A" followed by "B", "C", and the bottom is "Z". The black square shows that the network memorized the pattern in the value of α. At $\alpha = 1$, $\alpha = 2$, and $\alpha = 3$, there are black squares throughout the patterns, which

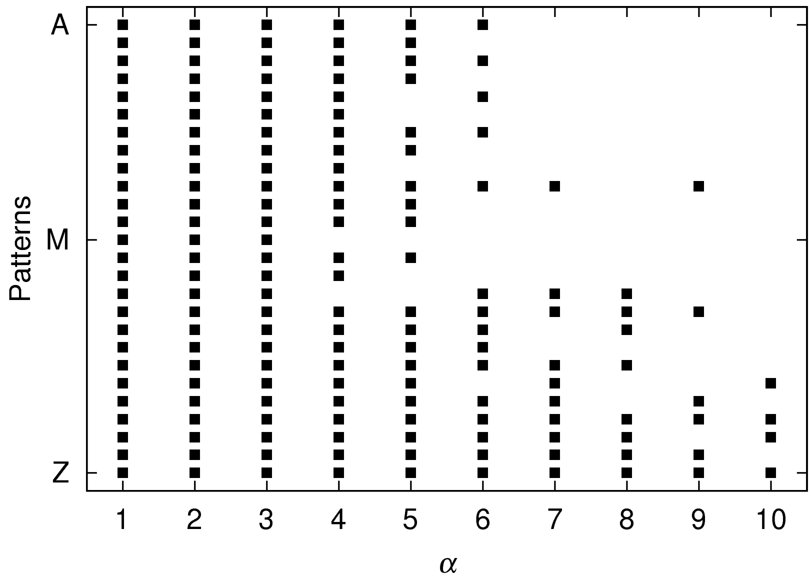
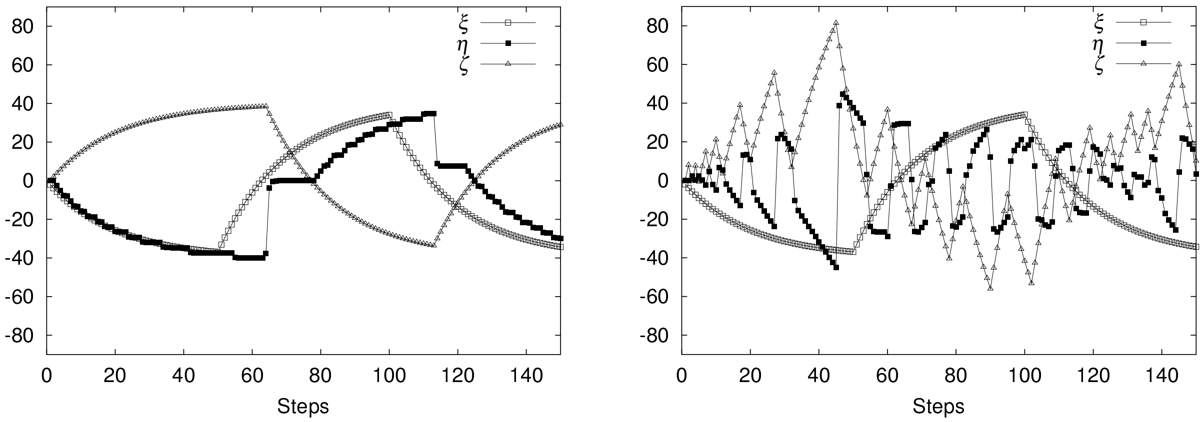


Figure 5: A pattern that the network learnt without noises is shown as a black square. In the high values of α, the head part, which is A, B, C, $\cdots$, are not learnt by the network.



(a) $\alpha = 2.0$

(b) $\alpha = 8.0$

Figure 6: Transitions of internal states of a neuron learning without noises. In (a), the mutual interaction η is following the external input ξ. In (b), η is not following η and is changing unstably.

means the network learnt all the patterns. At $\alpha = 7$, $\alpha = 8$, and $\alpha = 9$ the patterns figured "A" to "I" are not memorized in the network. At $\alpha = 10$, the patterns figured "A" to "T" are not memorized by the network. The patterns are inputted in alphabetical order. The head part of the input patterns go out of the memory of the network, being overwritten by the following patterns.

The other is that the learning is not effective to learn the patterns, because the internal states of the neurons are changing against our expectations. It seems to change unstable as shown in Figure 6. In both graphs, the horizontal axis shows the steps that the simulation continues and the vertical axis shows the values of the internal informations in a neuron. At $\alpha = 2.0$, after the external input changed at step 50, ξ is moving to positive. When ξ crosses 0, η and ζ start to change. After ζ crossed 0, the learning condition is satisfied again and again. At $\alpha = 8.0$, ζ changes its moving direction frequently. Therefore, the neurons change their connection weights right way and wrong alternately.

The verification of these hypothesizes remains for further works.

4 CONCLUSION

The incremental learning is another way to construct the associative memory, and can make the network learn from inputs with noises. In this paper, the computer simulations investigated how the refractoriness plays an important role in the incremental learning.

The results of the simulations show that the refractoriness is an essential factor. The neurons which have the refractoriness, like the chaotic neurons, must be used in this incremental learning. And they also show that the strong refractoriness causes failures of the learning of patterns. In this paper, two hypothetical reasons were offered, but the further detailed investigations remain for further works.

REFERENCES

[1] S. Asakawa, T. Deguchi, and N. Ishii, "On-Demand Learning in Neural Network," Proc. of the ACIS 2nd Intl. Conf. on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing, pp.84–89, 2001.

[2] M. Watanabe, K. Aihara, and S. Kondo, "Automatic learning in chaotic neural networks," Proc. of 1994 ieee symposium on emerging technologies and factory automation, pp.245–248, 1994.

[3] K. Aihara, T. Tanabe, and M. Toyoda, "Chaotic neural networks," Phys. Lett. A, Vol.144, No.6,7, pp.333–340, 1990.

[4] M. Watanabe, K. Aihara, and S. Kondo, "Self-organization dynamics in chaotic neural networks," Control and chaos, pp.320–333, Birkhauser, 1997.

[5] T. Ikeguchi, M. Adachi and K. Aihara, "Chaotic neural networks and associative memory," Proc. of Intl. Workshop on Artificial Neural Network, pp.17–24, 1991.

[6] Y. Osana, and M. Hagiwara, "Successive learning in hetero-associative memories using chaotic neural networks," Intl. Journal of Neural Systerms, vol.9, no.4, pp.285–299, 1999.

[7] T. Deguchi and N. Ishii, "Simulation results on the rate of success in chaotic search of patterns in neural networks," Intl. Journal of Chaos Theory and Applications, Vol.2, Nr.1, pp.47–57, 1997.

[8] T. Deguchi and N. Ishii, "Search of general patterns in the chaotic neural network by using pattern translation," Intl. Journal of Knowledge-Based Intelligent Engineering Systems, Vol.3, No.4, pp.205–214, 1999.