PAPER

Performance of Affordable Neural Network for Back Propagation Learning

Yoko UWATE[†a)], ***Student Member and*** **Yoshifumi NISHIO**[†b)], ***Member***

SUMMARY Cell assembly is one of explanations of information processing in the brain, in which an information is represented by a firing space pattern of a group of plural neurons. On the other hand, effectiveness of neural network has been confirmed in pattern recognition, system control, signal processing, and so on, since the back propagation learning was proposed. In this study, we propose a new network structure with affordable neurons in the hidden layer of the feedforward neural network. Computer simulated results show that the proposed network exhibits a good performance for the back propagation learning. Furthermore, we confirm the proposed network has a good generalization ability.

key words: *cell assembly, affordable neurons, back propagation*

1. Introduction

Recently, studies on the brain have been carried out actively on various levels. On the system level, many researchers investigate how the brain operates to realize higher functions such as learning, memory, emotion, and so on. Cell assembly [1]–[4] is one of explanations of information processing in the brain, in which an information is represented by a firing space pattern of a group of plural neurons. This mechanism has been proposed by Hebb who is a psychologist in 1949. Cell assembly could achieve multiple information processing carried out at the same time and parallel dispersion processing of the brain by forming a functional circuit with some neurons according to the necessary processing. Although this concept is impressive, we consider that more complex mechanism is realized in the brain.

On the other hand, Back Propagation (BP) learning [5] is one of engineering applications of neural networks. The BP learning operates with a feedforward neural network which is composed of an input layer, a hidden layer and an output layer, and the effectiveness of the BP learning has been confirmed in pattern recognition, system control, signal processing, and so on [6]–[8]. When the BP learning is applied, the hidden layer plays the essential role to decide the performance of the neural network. Hence, for example, there have been a lot of reports on the number of the neurons in the hidden layer. However, there have not been many studies on changing the structure of the hidden layer to improve the performance on convergence speed or learning efficiency.

In this study, we propose a new network model which is a feedforward neural network with affordable neurons in the hidden layer for more efficient BP learning. In this network, we prepare some extra neurons in the hidden layer. When the network executes the BP learning, all of the neurons in the hidden layer are not used at every updating. Namely, some of the neurons are selected for the learning and the rest of the neurons are deactivated. Computer simulated results show that the proposed network exhibits a good performance for the BP learning on both convergence speed and learning efficiency. Furthermore, we investigate a generalization ability of the proposed network by inputting an unknown pattern after learning. We are interested in such a generalization ability of artificial neural networks, because generalization should be one of the important mechanisms of higher functions in the brain.

2. Affordable Neural Network

Cell assembly is one of explanations of information processing in the brain, in which an information is represented by a firing space pattern of a group of plural neurons. This mechanism could achieve multiple information processing carried out at the same time and parallel dispersion processing of the brain by forming a functional circuit with some neurons according to the necessary processing. Conceptual diagram of cell assembly is shown in Fig. 1. Informations "*A*" and "*B*" are expressed by the corresponding functional connections between plural neurons. It is a special characteristics of cell assembly to exist overlapping neurons for the informations "*A*" and "*B*." Although this concept is impressive, we consider that more complex mechanism is realized in the brain. For example, in the case of cell assembly, an information "*A*" is expressed by a specific group of the neu-

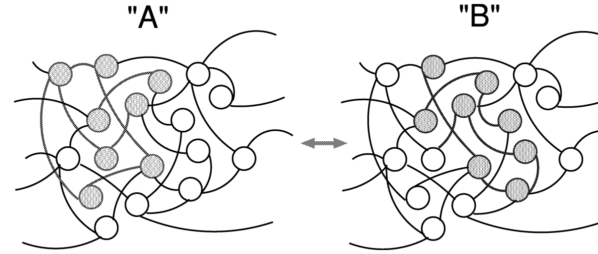
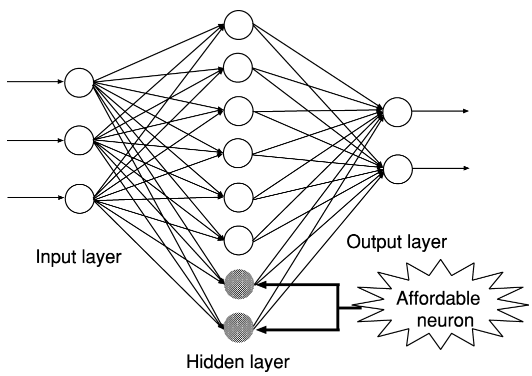


Fig. 1 Conceptual diagram of cell assembly.

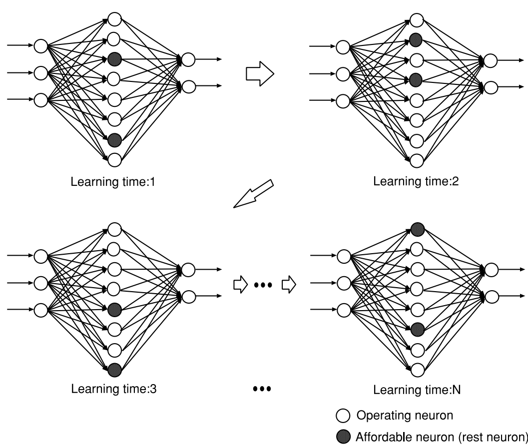
Manuscript received December 27, 2005.
Manuscript revised May 13, 2006.
Final manuscript received June 8, 2006.

[†]The authors are with the Department of Electrical and Electronic Engineering, Tokushima University, Tokushima-shi, 770-8506 Japan.
a) E-mail: uwate@ee.tokushima-u.ac.jp
b) E-mail: nishio@ee.tokushima-u.ac.jp

DOI: 10.1093/ietfec/e89–a.9.2374



(a) Network model with affordable neurons.



(b) Operation of the affordable neurons in the hidden layer.

Fig. 2 Affordable neural network.

rons. However, we consider that the same information "*A*" can be expressed by several different groups of neurons in the brain.

We introduce affordable neurons in the hidden layer of the feedforward neural network to reflect a function of the brain. The extra neurons in the hidden layer are prepared in advance. During the BP learning, all of the neurons in the hidden layer are not used at every updating. Namely, some of the neurons are selected for the learning and the rest of the neurons are deactivated. The network model with the affordable neurons is shown in Fig. 2.

For the proposed network, some of the neurons have to be selected at every input pattern. In this study, we consider the case that the affordable neurons in the hidden layer are selected at random during BP learning. One example of the temporal evolution under random selection is shown in Fig. 3. In this example, twenty neurons in the hidden layer are prepared and the number of the affordable neurons are set to two. The horizontal axis is time and the vertical axis is the neuron number in the hidden layer. The plotted marks indicate the neurons selected as affordable neurons at each updating. We can see that the affordable neurons are selected at random.

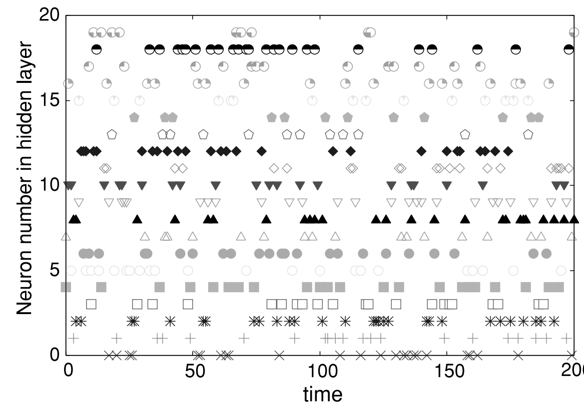


Fig. 3 Affordable neurons selected at random.

3. BP Learning Algorithm

The standard BP learning algorithm was introduced in [5]. The BP is the most common learning algorithm for feedforward neural networks. In this study, we use the batch BP learning algorithm. The batch BP learning algorithm is expressed by similar formula of the standard BP learning algorithm. The difference lies in the timing of the update of the weight. The update of the standard BP is performed after each single input data, while the update of the batch BP is performed after all different input data.

The total error E of the network is defined as the following equation.

$$E = \sum_{p=1}^{P} E_p = \sum_{p=1}^{P} \left\{ \frac{1}{2} \sum_{i=1}^{N} (t_{pi} - o_{pi})^2 \right\} \quad (1)$$

where P is the number of the input data, N is the number of the neurons in the output layer, t_{pi} denotes the value of the desired target data of the ith neuron for the pth input data, and o_{pi} denotes the value of the output data of the ith neuron for the pth input data. The goal of the learning is to set weights between all layers of the network to minimize the total error E. In order to minimize E, the weights are adjusted according to the following equation:

$$w_{i,j}^{k-1,k}(m+1) = w_{i,j}^{k-1,k}(m) + \sum_{p=1}^{P} \Delta_p w_{i,j}^{k-1,k}(m)$$
$$\Delta_p w_{i,j}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}} \quad (2)$$

where $w_{i,j}^{k-1,k}$ is the weight between the ith neuron of the layer $k-1$ and the jth neuron of the layer k, m is the learning time, and η is a proportionality factor known as the learning rate. In this study, we introduce inertia term in the 2nd term of the right-hand side of Eq. (2).

$$\Delta_p w_{i,j}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}} + \zeta \Delta_p w_{i,j}^{k-1,k}(m-1) \quad (3)$$

where ζ denotes the inertia rate.

4. Simulated Results

In this study, we investigate the performance of the affordable neural network for the two cases of the learning example.

4.1 Learning Example: $y = x^2$

First, we consider the feedforward neural network producing output x^2 for input data x as one learning example. The sampling range of the input data is $[-1.0,\ 1.0]$ and the step size of the input data is set to be 0.01. We carried out the BP learning by using the following parameters. The learning rate and inertia rate are fixed as $\eta = 0.1$ and $\zeta = 0.02$, respectively. Initial values of the weights are given between -1.0 and 1.0 at random. The learning time is set to be 50000. We investigate the convergence speed and the learning efficiency as the average of the total error between the output and the desired target. We define "Average Error E_{ave}" by the following equation.

$$E_{ave} = \frac{1}{P}\sum_{p=1}^{P}\left\{\frac{1}{2}(t_p - o_p)^2\right\}. \quad (4)$$

We prepare 8 neurons in the hidden layer and the number of the affordable neurons is set to 2. Namely, only 6 neurons are operated at every update time. This network is denoted as "affordable neural network (8-2)." For comparison, we investigate performance of the conventional neural networks without affordable neurons, when the number of neurons in the hidden layer is changed to 4, 6 and 8.

The simulation results are shown in Fig. 4. The horizontal axis is the learning time and the vertical axis is E_{ave}. We can see both of the convergence speed and the learning efficiency from the figure. In this figure, four graphs represent the results of the affordable neural network (8-2) and the conventional neural network (4-0), (6-0) and (8-0).

We can confirm that the improvement of the learning efficiency is difficult by only increasing the number of the neurons in the hidden layer. However, the proposed network with the affordable neurons gains a good performance. We consider that the network with the affordable neurons has an advantage to escape from local minima corresponding to some stable states.

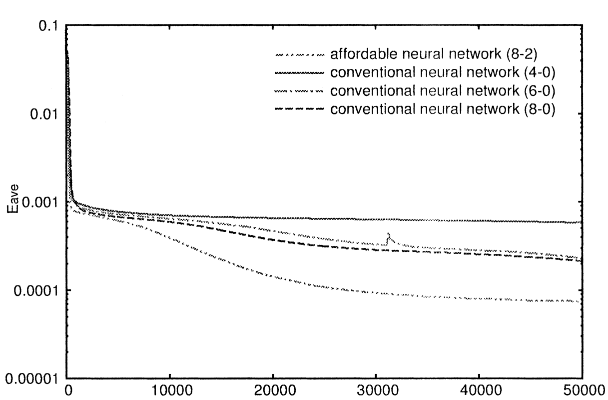
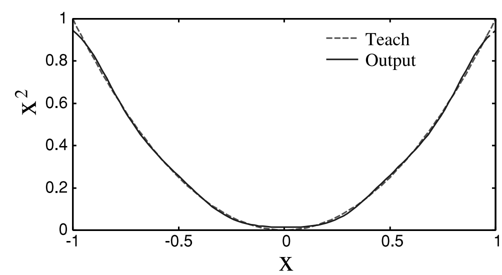


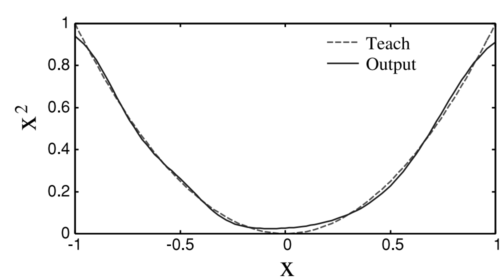
Fig. 4 Learning curve for different network structures.

Figure 5 shows the final results after the BP learning of Fig. 4. The simulated results of the affordable neural network (8-2) and the conventional neural network (8-0) are shown in Fig. 5 (a) and (b), respectively. The solid line denotes the output of the network after learning, and the dashed line denotes the teach data. From these figure, the case of conventional neural network, the error between output and teach data is large. However, the error of the affordable neural network is very small. We can see that the affordable neural network can learn the input data more precisely.

Next, the simulation results of the affordable neural network when the number of the affordable neurons is



(a) Affordable neural network (8-2).



(b) Conventional neural network (8-0).

Fig. 5 Results of learning for different network structures.

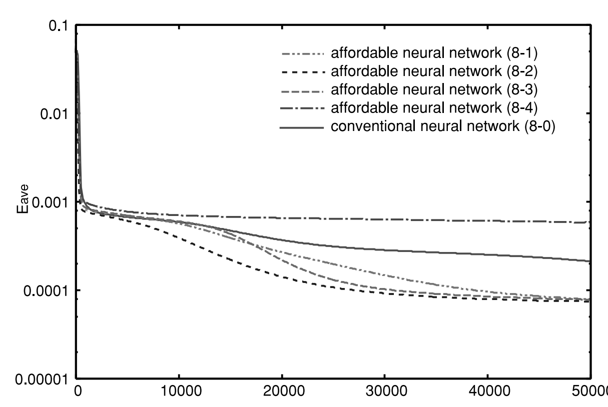
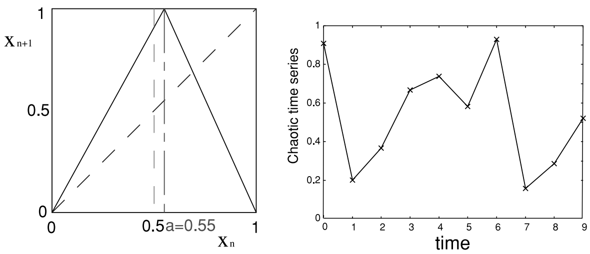


Fig. 6 Learning curve of the affordable neural network when the number of the affordable neurons are changed. (8 neurons are prepared in the hidden layer.)



(a) Skew tent map. (b) Time series.

Fig. 7 Skew tent map.

changed from 1 to 4 are shown in Fig. 6. When the number of the affordable neurons is set to 1 to 3, the network gain the better performance than the conventional neural network. However, the case of the number of the affordable neurons is set to 4, the network does not work well. We can confirm that the appropriate number of the affordable neuron is exist for given problems.

4.2 Learning Example: Chaotic Time Series

We consider that the affordable neural network has a generalization ability as well as a learning ability by virtue of its inefficient learning process. In this section, we investigate the learning ability and the generalization ability of the affordable neural network. We consider the learning of the structure of the skew tent map by training the network to output the same time series as the input time series produced by the skew tent map.

The skew tent map and an example of time series are shown in Fig. 7. The length of chaotic time series is set to 10 and the number of learning patterns is set to 100. When the network learns 10 lengths of time series, 10 nodes are prepared in the input and the output layers. Each data is inputted to each node in the input layer. We carried out the BP learning by using the following parameters. The parameter of the inertia rate is fixed as $\eta = 0.05$ and the initial values of the weights are given between -1.0 and 1.0 at random. The learning time is set to 20000.

4.2.1 Learning Ability

First, we investigate the learning efficiency as the average of the total error between the output and the desired target, when the network structure of the hidden layer is changed. The "Average Error E_{ave}" for this learning example is defined by the following equation.

$$E_{ave} = \frac{1}{P}\frac{1}{N}\sum_{p=1}^{P}\left\{\frac{1}{2}\sum_{i=1}^{N}(t_{pi} - o_{pi})^2\right\}. \quad (5)$$

We prepare 20 neurons in the hidden layer and the number of the affordable neurons is changed from 0 to 10. When the number of the affordable neurons is 0, the network has the normal (conventional) hidden layer structure. While, when the number of affordable neurons is 10, only 10 neurons of 20 neurons in the hidden layer operate at every update time.

The simulation results are shown in Fig. 8. The horizontal axis is the number of the affordable neurons and the vertical axis is E_{ave} for the learning pattern. Where E_{ave} denotes average ten simulations. In this figure, we confirm that the neural network with the affordable neurons gains better performance than the conventional neural network. However, when the number of the affordable neurons exceeds 4, the performance of the network gradually becomes worse because the number of the operating neurons decreases and the network becomes sluggish.

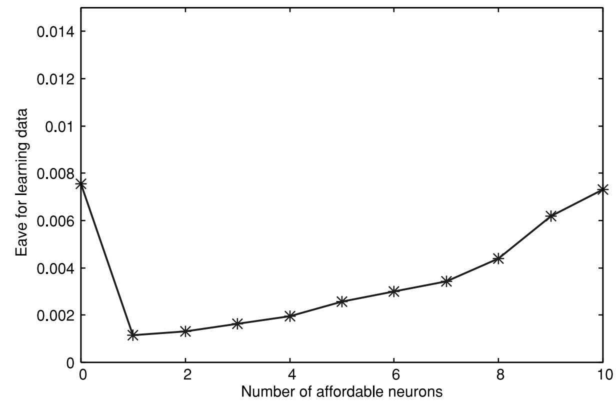


Fig. 8 E_{ave} for learning data.

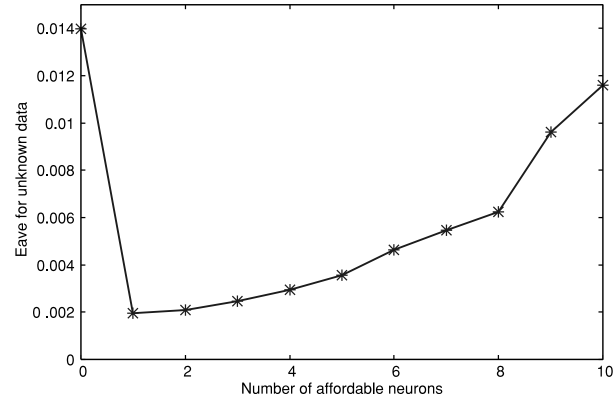


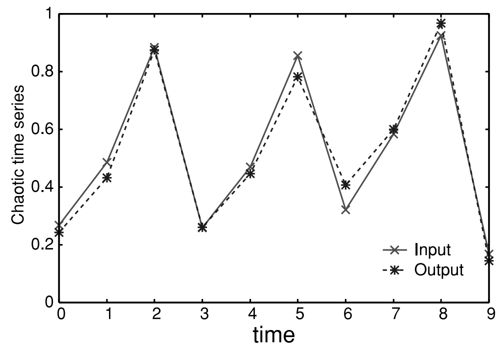
Fig. 9 E_{ave} for unknown data.

4.2.2 Generalization Ability

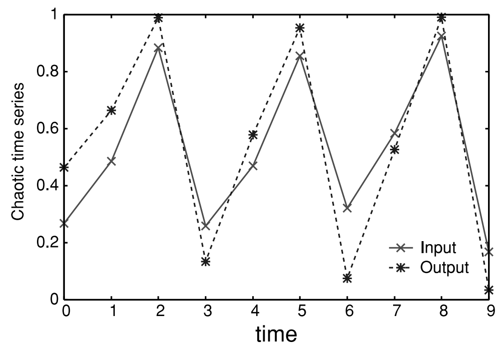
Next, we investigate a generalization ability of the affordable neural network. After the above learning of 100 patterns of the time series, we input an unknown chaotic time series generated the same skew tent map as an input pattern.

The simulation result for unknown chaotic time series after learning of the network is shown in Fig. 9. The horizontal axis is the number of the affordable neurons and the vertical axis is E_{ave} for 10 different unknown input data. In this figure, the affordable neural network gains good performance as well as the above case of the learning data.

One example of results for unknown chaotic time series is shown in Fig. 10. The horizontal axis is time and the ver-

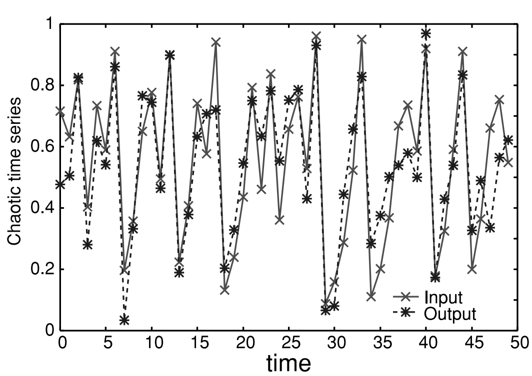


(a) Affordable neural network (20-2). (E_{ave}: 0.00129).

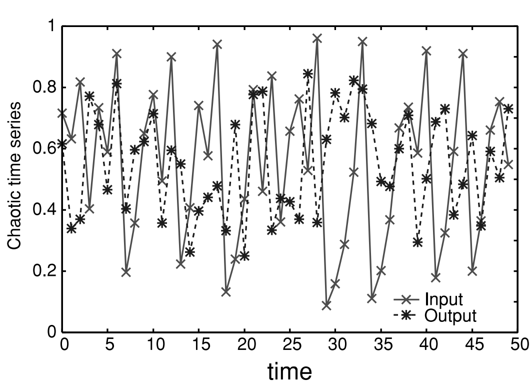


(b) Conventional neural network (20-0). (E_{ave}: 0.01023).

Fig. 10 Comparison between unknown input chaotic time series and output of the neural networks (Length of chaotic time series: 10).



(a) Affordable neural network (100-15). (E_{ave}: 0.00737).



(b) Conventional neural network (100-0). (E_{ave}: 0.05059).

Fig. 11 Comparison between unknown input chaotic time series and output of the neural networks (Length of chaotic time series: 50).

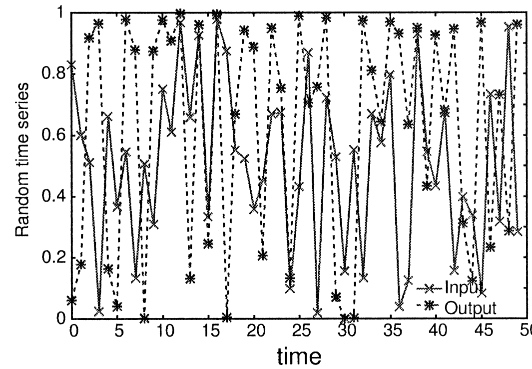


Fig. 12 Comparison between unknown random input data and output of the affordable neural networks (100-15) (Length of input data: 50, E_{ave}: 0.11640).

tical axis is unknown chaotic time series generated by the skew tent map. The affordable neural network almost can output the same unknown input data (Fig. 10(a)). The performance of the conventional neural network is worse for unknown input data (Fig. 10(b)).

Furthermore, Fig. 11 shows one example of results when the length of chaotic time series is 50. We prepare 100 neurons in the hidden layer. Figure 11(a) shows the case that the number of the affordable neurons is 15. Figure 11(b) shows the case of the network without affordable neurons in the hidden layer. From these figures, we can confirm that the affordable neural network gains a good performance to reproduce unknown chaotic time series.

Before concluding that the affordable neural network can learn the structure of the skew tent map from the inputted time series, we have to confirm that the network does not learn to output the same data as input data directly. We investigate the performance of the affordable neural network when a random data is inputted as unknown data after the learning. One example of the simulated results is shown in Fig. 12. The E_{ave} between the random input and output of the network is very large (E_{ave}: 0.11640). From this result, we can say that the affordable neural network does not output the same data as input data directly.

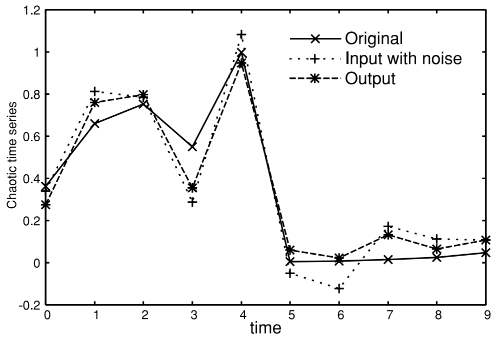
The affordable neural network does not learn efficiently. We consider that it is important not to learn efficiently for producing the generalization ability. In this example, the affordable neural network does not learn only the given data set but also some characteristics or features of the given data.

4.2.3 Noise Reduction

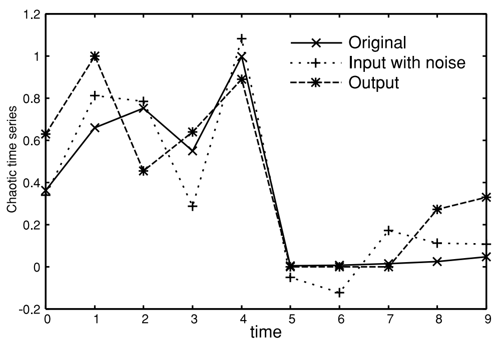
Finally, we consider the noise reduction by the affordable neural network. After the learning of 100 patterns of the time series generated the skew tent map, we input an unknown chaotic time series generated by the same map polluted by uniform noise. The length of chaotic time series is set to 10. The range of the uniform noise is set to $[-0.3, 0.3]$. We define the "Noise Reduction Error E_{NR}" to evaluate noise reduction as the following equation.

Table 1 E_{NR} for the noise reduction.

Affordable NN	Conventional NN
−0.11158	0.26582



(a) Affordable neural network (20-2). (E_{NR}: −0.06189).



(b) Conventional neural network (20-0). (E_{NR}: 0.28017).

Fig. 13 Results of noise reduction.

$$E_{NR} = \sum_{i=1}^{N} (\hat{t}_i - o_i)^2 - \sum_{i=1}^{N} (\hat{t}_i - t_i)^2, \qquad (6)$$

where $\hat{t}_i$ denotes the original data, t_i is the input data with noise and o_i is the output data.

The average of 20 simulated results are summarized in Table 1 and one example of simulated results of the noise reduction is shown in Fig. 13. The solid line is the original data, the dotted line is the input data with noise, and the dashed line is the output data. From these figures, we confirm the affordable neural network can reduce the noise added to the original data. However, the conventional network does not operate for the noise reduction.

5. Conclusions

In this study, we have proposed a new network model which was a feedforward neural network with affordable neurons in the hidden layer for more efficient BP learning. Computer simulated results showed that the proposed network with affordable neurons gained a good performance for the BP learning on both convergence speed and learning efficiency. Furthermore, we confirmed the affordable neural network has a good generalization ability.

In future work, we investigate various kinds of methods to select the affordable neurons in the hidden layer. because chaotic selection methods might be efficient, This is because, we have confirmed that chaotic behavior can obtain good performance to escape out of local minima for combinatorial optimization problems [9]–[11].

References

[1] Y. Sakurai, "Dependence of functional synaptic connections of hippocampal and neocortical neurons on types of memory," Neuroscience Letters, vol.158, pp.181–184, 1993.

[2] E. Vaadia, I. Haalman, M. Abeles, H. Bergman, Y. Prut, H. Slovin, and A. Aertsen, "Dynamics of neural interactions in monkey cortex in relation to behavioral events," Nature, vol.373, pp.515–518, Feb. 1995.

[3] Y. Sakurai, "Hippocampal and neocortical cell assemblies encode memory processes for different types of stimuli in the rat," Journal of Neuroscience, vol.16, pp.2809–2819, 1996.

[4] Y. Sakurai, "How do cell assemblies encode information in the brain?," Neuroscience and Biobehavioral Reviews, vol.23, pp.785–796, 1999.

[5] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representations by error propagation," in Parallel Distributed Processing, ed. D.E. Rumelhart and J.L. McCelland, vol.1, pp.318–362, MIT Press, MA, 1986.

[6] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, "Handwritten digit recognition with a back-propagation network," in Advances in Neural Information Processing Systems, ed. D.S. Touretzky, vol.2, MIT Press, MA, 1990.

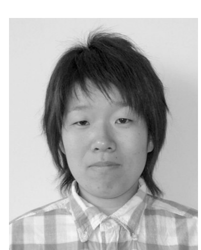
[7] M. Kawano, Y. Uno, M. Isobe, and R. Suzuki, "A hierarchical neural network model for voluntary movement with application to robotics," IEEE Control Syst. Mag., vol.8, no.2, pp.8–16, 1988.

[8] T.J. Sejnowski and C.R. Rosenberg, "Parallel networks that learn to pronounce English text," Complex Systems, vol.1, pp.145–168, 1987.

[9] T. Ueta, Y. Nishio, and T. Kawabe, "Comparison between chaotic noise and burst noise on solving ability of Hopfield neural networks," Proc. NOLTA'97, vol.1, pp.409–412, Nov. 1997.

[10] Y. Uwate, Y. Nishio, and A. Ushida, "Markov chain modeling of intermittency chaos and its application to Hopfield NN," IEICE Trans. Fundamentals, vol.E87-A, no.4, pp.774–779, April 2004.

[11] Y. Uwate, Y. Nishio, T. Ueta, T. Kawabe, and T. Ikeguchi, "Performance of chaos and burst noise injected to the Hopfield NN for quadratic assignment problems," IEICE Trans. Fundamentals, vol.E87-A, no.4, pp.937–943, April 2004.



Yoko Uwate was born in Tokushima, Japan, in 1980. She received the B.E. and M.E. degrees in Electrical and Electronic Engineering from Tokushima University, Tokushima, Japan, in 2003 and 2005, respectively. She is currently working towards her Ph.D. degree at the same university. Her research interests include complex phenomena in chaotic circuits and neural networks. Ms. Uwate is a student member of the IEEE.

Yoshifumi Nishio received B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama Japan, in 1988, 1990, and 1993, respectively. In 1993, he joined the Department of Electrical and Electronic Engineering at Tokushima University, Tokushima Japan, where he is currently an Associate Professor. From May 2000 he spent a year in the Laboratory of Nonlinear Systems (LANOS) at the Swiss Federal Institute of Technology Lausanne (EPFL) as a Visiting Professor. His research interests include analysis and application of chaos in electrical circuits, analysis of synchronization in coupled oscillatory circuits, development of analyzing methods for nonlinear circuits, theory and application of cellular neural networks, and neural network architecture. He was the Chair of the IEEE CAS Technical Committee on Nonlinear Circuits and Systems (NCAS) during 2004–2005 and is currently the Steering Committee Secretary of the IEICE Research Society of Nonlinear Theory and its Applications (NOLTA). He is serving as an Associate Editor for the IEEE Transactions on Circuits and Systems Part I and the RISP Journal of Signal Processing. He is a member of the IEEE, and the RISP.