第19回　回路とシステム
軽井沢ワークショップ
The 19th Workshop on Circuits and Systems
in Karuizawa, April 24-25, 2006

Durability of Affordable Neural Networks during Back Propagation Learning

Yoko Uwate† , Yoshifumi Nishio† and Ruedi Stoop‡
† Dept. of Electrical and Electronics Engineering, Tokushima University
2-1 Minami-Josanjima, Tokushima, Japan
Email: {uwate, nishio}@ee.tokushima-u.ac.jp
‡ ‡ Institute of Neuroinfomatics, University / ETH Zurich
Winterthurerstrasse 190, 8057 Zurich, Switzerland
Email: ruedi@ini.phys.ethz.ch

1 Introduction

Recently, studies on the brain have been carried out actively on various levels. On the neuroscience level, many researchers investigate how neuronal death occurs in the human brain. Researches of neuronal death are help to make clear the mechanisms of unrevealed diseases such as Alzheimer disease [1]. Neuronal death can be occurred by various but usual causes such as drinking a lot of alcohol, heavy smoking and concussions. Furthermore, some neurons die with apoptosis when the neuron can not receive any signals.

Some neurons often die in the human brain in daily life. Even if some neurons die, the human brain is still able to operate normally by creating new information processing from abundant experience and knowledge. This is because, the human brain has durability and flexibility.

Many modelings of the human brain with the visual or the audio sensation are reported [2]-[4] due to development of the brain researches. However, the investigation of modeling of higher functions in the human brain such as durability and flexibility is just getting started. We consider that it is very important to apply these high functional mechanisms of the human brain to novel artificial neural networks.

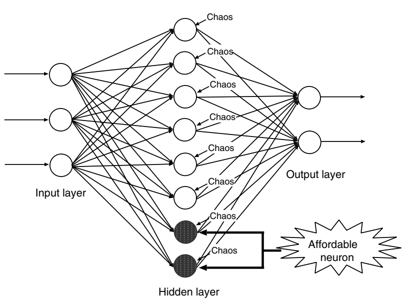
In our previous research, we have proposed a new network structure with affordable neurons in the hidden layer, for efficient BP learning [5][6]. We named this network “Affordable Neural Network.” In this network, we prepare some extra neurons in the hidden layer. When the network executes operating, all of the neurons in the hidden layer are not used at every updating. Namely, this network is able to operate with high function by using different neurons in the hidden layer. By computer simulations, the affordable neural network has been confirmed to gain better performance for the BP learning on both convergence speed and learning efficiency. Further, we have investigated the performance of the affordable neural network when noise-polluted data is inputted. And we have confirmed that the affordable neural network is able to output noise-cleaned data. We can say that the affordable neural network has generalization ability. However, we believe that many advantages of the affordable neural network are still veiled. We consider that the affordable neural network is able to self-generate network durability.

In this study, we investigate the durability of the affordable neural network when some of the neurons in the hidden layer are damaged, after the learning process. If a neuron is selected as a damaging neuron, the connections to the output layer from the neuron are cut. Namely, the damaging neuron does not operate. By computer simulations, we confirm that the affordable neural network keeps its efficiency. We conclude that the affordable neurons exert an important influence on durability of the network.

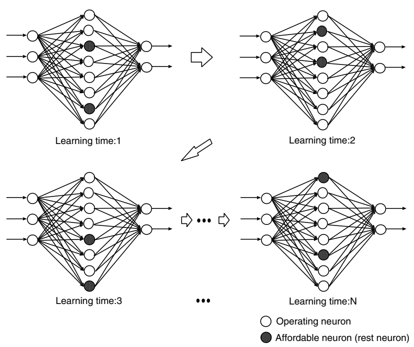
2 Affordable Neural Network

2.1 Network Model with Affordable Neurons

In Refs. [5][6], we have proposed a network with affordable neurons in the hidden layer of the feedforward neural network structure for efficient BP learning. We introduced the affordable neurons to reflect important properties of the brain. During the BP learning, not 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 and the operation of the affordable neurons in the hidden layer are shown in Fig. 1(a) and Fig. 1(b), respectively.



(a) Network model with affordable neurons.



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

Figure 1: Affordable neural network.

2.2 Chaotic Selection

For the proposed network, at every update, some of the neurons have to be selected for the BP learning. In Refs. [7]-[9], the authors have investigated the performance of the Hopfield neural network solving combinatorial optimization problems when chaos is inputted to the neurons as noise. By computer simulations, chaotic noise has been confirmed to improve performance by being able to escape from local minima much better than random noise. Hence, we consider that various features of chaos are effective for neural networks.

In this study, we use the skew tent map as a simple chaotic map, to realize a chaotic selection of operating neurons in the hidden layer. We prepare the skew tent maps with different initial values, whose number is the same as the number of the neurons in the hidden layer, so that each skew tent map corresponds to one neuron. At every update of the BP learning, the skew tent maps are also updated, and their values are referred. By selecting a number of neurons in the order of the values of their skew tent maps, the selection is made in a chaotic manner. Note that chaos is not inputted to the neural network directly, but is used only for the selection of the operating neurons. The skew tent map is defined by the following equation.

$$x_{n+1} = \begin{cases} \dfrac{2x_n + 1 - a}{1 + a} & (-1 \le x_n \le a) \\ \dfrac{-2x_n + 1 + a}{1 - a} & (a < x_n \le 1). \end{cases} \tag{1}$$

One example of the temporal evolution under chaos selection is shown in Fig. 2. In this example, eight 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 chaotically.

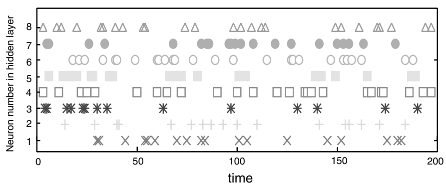


Figure 2: Chaotic selection.

3 BP Learning Algorithm

The standard BP learning algorithm was introduced in [10]. 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 a formula similar to the standard BP learning algorithm. The difference lies in the timing of the weight. The update of the standard BP is performed after each single input data, while for the batch BP the update is performed after all input data has been processed. The total error E of the network is defined as

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

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 for the pth input data, and o_{pi} denotes the value of the output data for the pth input data. The goal of the learning is to set weights between all layers of the network so as to minimize the total error E. In order to minimize E, the weights are adjusted according to the following equation:

$$w_{i,j}^{k-1,k}(m+1) = w_{i,j}^{k-1,k}(m) + \sum_{p=1}^{P} \Delta_p w_{i,j}^{k-1,k}(m),$$
$$\Delta_p w_{i,j}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}}, \qquad (3)$$

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 add to the second line of Eq.(3) an inertia term, leading to

$$\Delta_p w_{i,j}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}} + \zeta \Delta_p w_{i,j}^{k-1,k}(m-1), \qquad (4)$$

where ζ denotes the inertia rate.

4 Damaging Neurons

Even if some neurons of the human brain die, it is still able to operate normally in most cases. This durability is one of the most important features of the brain. In the following, we numerically investigate the durability of the affordable neural network.

We assume that some neurons in the hidden layer are damaged by some causes after the BP learning. The connections to the output layer of the damaged neurons are cut as shown in Fig. 3. Namely, the damaged neurons do not operate. In this situation, we investigate the performance of the network when the learning data are inputted to the network.

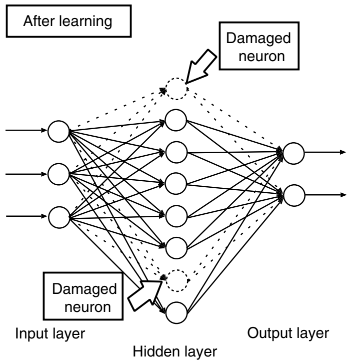


Figure 3: Damaging neurons.

5 Simulated Results

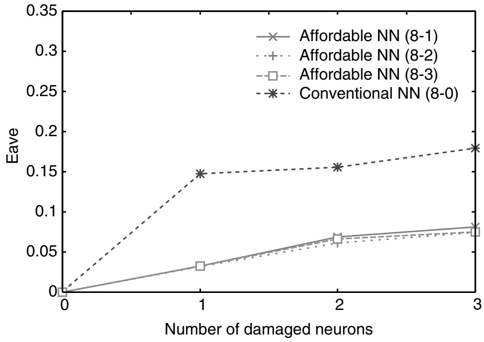
We consider the feedforward neural network for the task to produce output x^2 for input x, as one learning example. The sampling range of the input data is $[-1.0,\ 1.0]$ and the step size of the input data is set to be 0.01. Our BP learning was based on the fallowing parameters. The learning rate and the inertia rate are fixed as $\eta = 0.1$ and $\zeta = 0.02$, respectively, and initial values of the weights are given between -1.0 and 1.0 at random. The learning time is set to $m = 20000$. We investigate the total error between the output and the desired target when some neurons are damaged after learning. We define "Average Error E_{ave}" by the following equation.

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

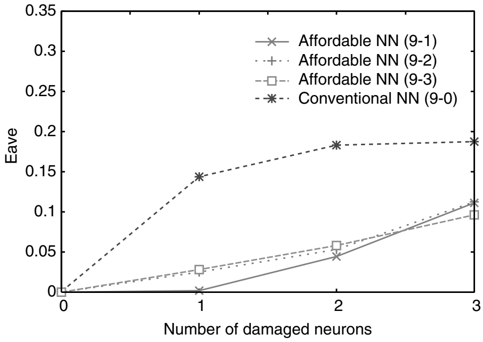
In this study, we prepare from 8 to 11 neurons in the hidden layer of the network and the number of the affordable neurons is set to 1 to 3. For example, the number of neurons in the hidden layer is 8 and the number of the affordable neurons is set to 2. In this case, only 6 neurons are operated every time. This network is denoted as "affordable neural network (8-2)." For comparison, we investigate the performance of the conventional neural network without any affordable neurons.

5.1 Effect of Damaged Neurons

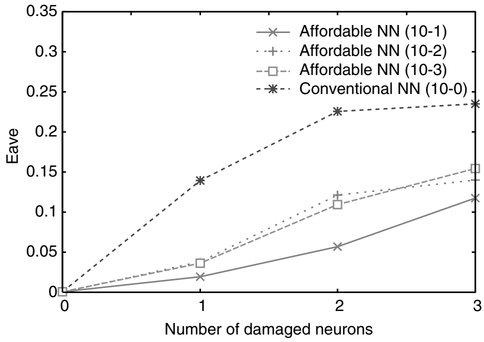
The simulation results of the affordable neural network and the conventional neural network



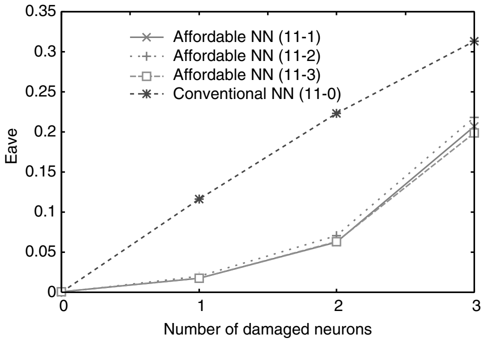
(a) Hidden layer: 8.



(b) Hidden layer: 9.



(c) Hidden layer: 10.



(d) Hidden layer: 11.

Figure 4: E_{ave} when some neurons are damaged.

when some neurons in the hidden layer are damaged after learning are shown in Fig. 4. Figures 4 (a), (b), (c) and (d) show the simulation results for the cases that the numbers of the neurons in the hidden layer are prepared as 8, 9, 10 and 11, respectively. The horizontal axis is the number of the damaged neuron in the hidden layer and the vertical axis is the average of E_{ave} for all combination of the damaged neurons. The E_{ave} of both the affordable neural network and the conventional neural network becomes worse by increasing the number of the damaged neurons. By comparison of two networks, the affordable neural networks gain better performance than the conventional neural network when some neurons are damaged. Because the E_{ave} of the affordable neural networks is about half the conventional neural networks. From these results, we confirmed that the affordable neural networks can operate well even if some neurons are damaged.

Some examples of outputs of the affordable neural network and the conventional neural network before and after some neurons are damaged are shown in Figs. 5 and 6. The left figure shows the output before some neurons are damaged. The right figure shows the output after some neurons are damaged. In the case of the affordable neural network, the effect of damaging neuron is small and the network gains good performance as before. While the case of the conventional neural network, the effect of damaging neuron is very large and the network does not operate well.

We consider that the learning process using different patterns of operating neurons in the affordable neural network prevents the neurons playing fixed roles and creates the durability of the whole network.

5.2 Position of damaging neurons

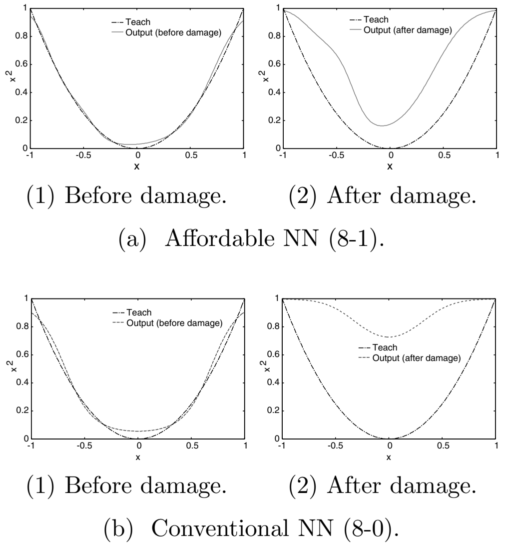
Next, we investigate the relationship between performance and position of the damaging neurons. The simulated results when the damaged neurons is only one are shown in Fig. 7. The horizontal axis is the position of the damaged neuron in the hidden layer and the vertical axis is E_{ave}. The solid line denotes the affordable neural network and the dashed line denotes the conventional neural network. We emphasize that the vertical axis in Fig. 7 is denoted in log scale. From these figures, E_{ave} of the conventional neural network is not good. And the performance of the conventional neural network does not depend on the position of the damaged neurons. On the other hand, E_{ave} of the affordable neural network is almost very small when any neuron in the hidden layer is damaged. For example, the case of the number of the neurons in the hidden layer is set to 11 (Fig. 7 (d)), the E_{ave} of the affordable neural network is less than one tenth or 1/100 of the E_{ave} of the conventional neural network for many positions.

We confirm that the conventional neural network is not able to operate when any neuron is damaged. However, the affordable neural network keeps the same performance as the network before some neurons damaged. This is because, the affordable neural network generates durability during BP learning.

6 Conclusions

In this study, we investigated the durability of the affordable neural network when some of the neurons in the hidden layer are damaged after the learning process. By computer simulations, we confirmed that the affordable neural network keeps its good performance. It is obvious that the affordable neurons exert an important influence on durability of the network. Further, we investigated the performance of chaos, regular and random selection methods of affordable neurons and found superior behavior of the chaotic and random choice over regular choice.

We believe thus the chaotic selection exerts an important beneficial effect on the network's learning in the hidden layer. In future work, we will investigate the efficacy of chaotic selection for more difficult problems in more complicated situations.



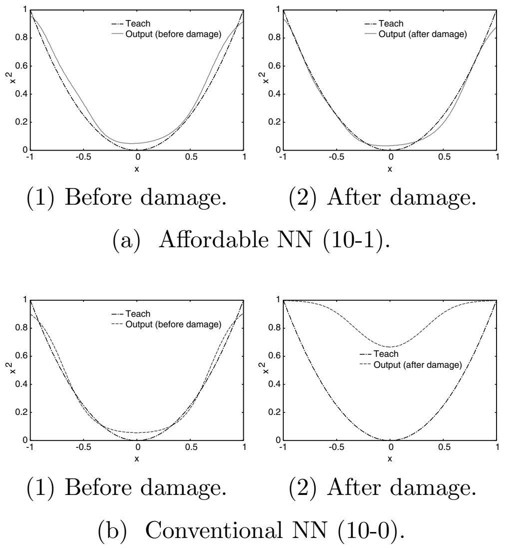
(1) Before damage. (2) After damage.

(a) Affordable NN (8-1).

(1) Before damage. (2) After damage.

(b) Conventional NN (8-0).

Figure 5: Output of the network before and after one neuron is damaged (Hidden: 8).



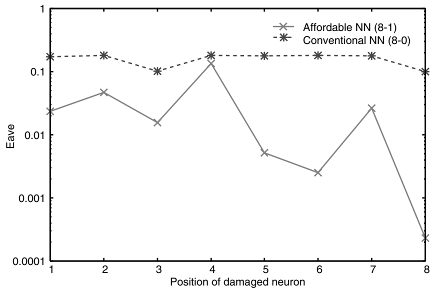
(1) Before damage. (2) After damage.

(a) Affordable NN (10-1).

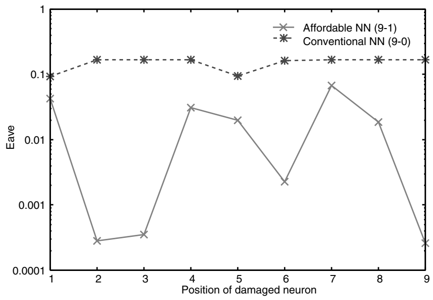
(1) Before damage. (2) After damage.

(b) Conventional NN (10-0).

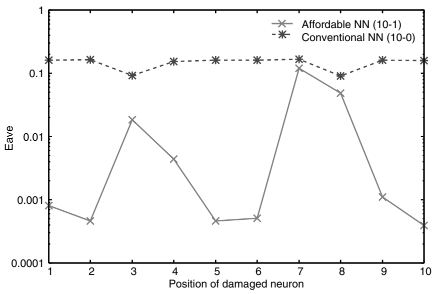
Figure 6: Output of the network before and after one neuron is damaged (Hidden: 10).



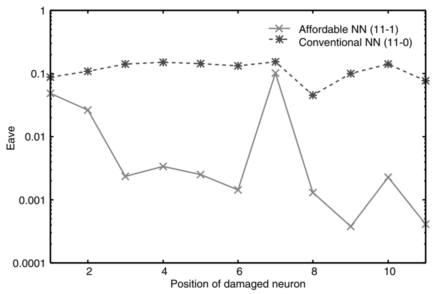
(a) Hidden layer: 8.



(b) Hidden layer: 9.



(c) Hidden layer: 10.



(d) Hidden layer: 11.

Figure 7: E_{ave} and position of damaged neurons.

References

[1] H. Satoi, H. Tomimoto, R. Ohtani, T. Kitano, T. Kondo, M. Watanabe, N. Oka, I. Akiguchi, S. Furuta, Y. Hirabayashi and T. Okazaki, "Astroglial Expression of Ceramide in Alzheimer's Disease Brains: A Role During Neuronal Apoptosis," Neuroscience, vol. 130, pp. 657-666, 2005.

[2] R. P. Broussard, S. K. Rogers, M. E. Oxley and G. L. Tarr, "Physiologically Motivated Image Fusion for Object Detection Using a Pulse Coupled Neural Network," IEEE Transactions on Neural Networks, 10 (3), pp. 554-563, 1999.

[3] R. Stoop and A. Kern, "Two-Tone Suppression and Combination Tone Generation as Computations Performed by the Hopf Cochlea," Phys. Rev. Lett., 93 (26) 268103, 2004.

[4] R. Stoop, W.-H. Steeb, J.C. Gallas and A. Kern, "Auditory Two-Tone Suppression from a Subcritical Hopf Cochlea," Physica A, 351: pp. 175-183, 2005.

[5] Y. Uwate and Y. Nishio, "Back Propagation Learning of an Affordable Neural Network for Chaotic Time Series," *Proceedings of RISP International Workshop on Nonlinear Circuits and Signal Processing (NCSP'05)*, pp. 139-142, Mar. 2005.

[6] Y. Uwate and Y. Nishio, "Back Propagation Learning of Neural Networks with Chaotically-Selected Affordable Neurons," *IEEE International Symposium on Circuits and Systems (ISCAS'05)*, pp. 1481-1484, May. 2005.

[7] 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.

[8] Y. Uwate, Y. Nishio and A. Ushida, "Markov Chain Modeling of Intermittency Chaos and its Application to Hopfield NN," *IEICE Transactions on Fundamentals*, vol. E87-A, no. 4, pp. 774-779, Apr. 2004.

[9] 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 Transactions on Fundamentals*, vol. E87-A, no. 4, pp. 937-943, Apr. 2004.

[10] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, vol. 1, MIT Press, MA, pp. 318-362, 1986.