2013 International Symposium on Nonlinear Theory and its Applications
NOLTA2013, Santa Fe, USA, September 8-11, 2013

Investigation of Influences of Neurogenesis in Multi-Layer Perceptron

Yuta Yokoyama†, Chihiro Ikuta†, Yoko Uwate† and Yoshifumi Nishio†

†Department of Electrical and Electronic Engineering, Tokushima University
2-1 Minami-Josanjima, Tokushima-shi, Tokushima, 770-8506, Japan
Email: {yuta, ikuta, uwate, nishio}@ee.tokushima-u.ac.jp

Abstract—Neurogenesis is that new neurons are generated in the human brain. We focus on the characteristic of the neurogenesis with biologically. In the previous study, we have proposed artificial network model which was applied the neurogenesis to Multi-Layer Perceptron (MLP).

In this study, we show the effectiveness of the proposed network with neurogenesis for pattern recognition. And we investigate the parameter dependency for detailed research on the influences of neurogenesis on MLP.

1. Introduction

It is said that there are about 10 billion neurons in the human's brain. The network is formed by connecting of more than one neuron. However, neurons had been considered to be lost with age until several years ago. In recent studies, some researchers reported that new neurons are generated in the dentate gyrus of hippocumpus [1]- [3]. This process is called "neurogenesis". By utilizing the neurogenesis, some brain cells increase and the network is substantial. It is known that the neurogenesis improves ability to solve problems by combining new neurons. We focus on the characteristic of the neurogenesis on biology. In the previous study, we have proposed artificial network model which was applied the neurogenesis to Recurrent Neural Network (RNN) [4] and Multi-Layer Perceptron (MLP) [5]. Therefore, we have researched the performance of the proposed network.

In this study, we investigate the parameter dependency for detailed research on the influences of neurogenesis on MLP. Then, we research the learning performance of the proposed network and focus on the some parameters of the neurogenesis using the proposed network.

2. MLP with Neurogenesis

2.1. Neurogenesis

Before, there is no neural stem cell which makes a neuron in the brain of an adult. It was impossible to generate the new neuron. Therefore, the neurogenesis had been considered to generate for period of growth. Thus, neurons had been considered to be lost with age until several years ago. However, some researchers reported that new neurons are generated in the adult brain. This process is called "neurogenesis". The neurogenesis in the hippocumpus of the human brain was discovered in the late 1990s by Erickson et al [1]-[3]. The neurogenesis is included in the existing neural circuit by given the learning and new neurons are generated in the human brain. It is known that the neurogenesis improves memory, learning, thinking ability, and so on. We focus attention on the characteristic of the neurogenesis.

2.2. Rule of Neuron Updating

We use a Multi-Layer Perceptron (MLP) which is comparatively easy network for artificial neural network and one of a feed-forward neural network. The MLP is a most famous feed-forward neural network. This network is used for pattern recognition, time series prediction, noise reduction, motion control, and other tasks. The MLP is composed some layers of neuron, it has input layer, hidden layer, and output layer. This network learns to the tasks by changing the weight parameters. Generally, the performance of the MLP is changed by the number of neurons. Moreover, we used the Back Propagation (BP) which is one of the MLP's learning method.

A Back Propagation (BP) is used to the MLP's learning algorithm. The BP was introduced by D.E. Rumelhart in 1986 [6]-[8]. In this algorithm, the network calculate the error from the output and teaching signal.

The neuron has the multi inputs and a single output. The updating rule of neuron is described by Eq. (1).

$$x_i(t+1) = f\left(\sum_{j}^{n} w_{ij}(t)x_j(t) - \theta\right), \qquad (1)$$

where x is the input or output and w is the connection weight parameter and θ is threshold. In this equation, the weight of connection and threshold of neuron are learned by BP algorithm. We used the sigmoid function for the output function. The error of MLP propagates backward in the feed-forward neural network. BP algorithm changes value of weights to obtain smaller error than before. The total error E of the network is described by Eq. (2).

$$E = \frac{1}{2}\sum_{p=1}^{p}\sum_{i=1}^{n}(t_{pi} - o_{pi})^2, \qquad (2)$$

where E is the error value, p is the number of the input data, n is the number of the neurons in the output layer, t_{pi} is the value of the desired target data for the pth input data, and o_{pi} is the value of the output data for the pth input data. The function of the connection weight is described by Eq. (3).

$$\Delta_p w_{ij}^{k-1,k} = \eta_{pj}^{k} o_{pi}^{k-1} = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}}, \quad (3)$$

where $w_{i,j}^{k-1,k}$ is the weight between the ith neuron of the layer $k-1$ and the j the neuron of the layer k, and η is the proportionality factor known as the learning rate.

2.3. Network structure

In this study, we apply the behavior of neurogenesis to the MLP. We have proposed artificial network model which was applied the neurogenesis to MLP [5]. Figure 1 shows a structure of the proposed network.

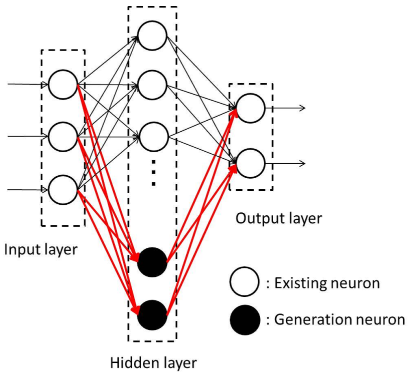

Figure 1: MLP with neurogenesis.

In fact, the timing of generation new neurons are less certain. Then, we consider two kinds of the neurogenesis. One is the periodic neurogenesis. In the case of the periodic neurogenesis, new neurons generated at every 10 iterations during the learning process. The other is the chaotic neurogenesis. In the case of the chaotic neurogenesis, new neurons generated by using the logistic map. Hasegawa et al. investigated solving abilities of the Hop-field NN with various surrogate noise, and they concluded that the effects of the chaotic sequence for solving optimization problems can be replaced by stochastic noise with similar autocorrelation [9]. The updating function of the logistic map is described by Eq. (4).

$$y(n+1) = \alpha y(n)(1-y(n)). \quad (4)$$

If the parameter α is changed, Eq. (4) will be served in chaos through period-doubling bifurcation. In this study, we use that the parameter α = 3.82740 and 4.0. When we choose that the parameter α is 3.82740, it is well known that the map produces intermittent bursts just before periodic-windows. In cases where the parameter α is 3.82740, it is the intermittently chaotic time series. This time series could be divided into two phases (laminar part of periodic behavior with period 3 and burst part of spread points over the invariant interval). Moreover, in cases where the parameter α is 4.0, it is the fully chaotic time series. In this study, we consider y = 0.6 as basis. In this timing method, the neurogenesis is occurred when the value of y takes the range between 0.6 to 0.7.

3. Simulations

We consider a pattern recognition, where 26 alphabets patterns consisting of 0 and 1 in Fig. 2 are stored in the neural network for recognition.

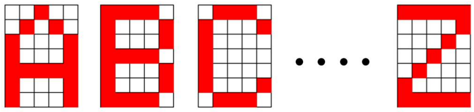

Figure 2: Input patterns.

We consider that the propose network is composed of three layers. The number of neurons in the input layer is 35, and the output layer is 26. In the proposed MLPs, we set 10 neurons in the hidden layer at the start of learning. Therefore, we set the maximum number of the neurons in the hidden layer. The MLPs learn 5000 times during the one trial. The learning rate is $\eta = 0.005$ and initial values of the weights are between -0.5 and 0.5 at random. Moreover, we compare the learning performance of the following five kinds of MLPs:

1. The conventional MLP
2. The MLP with random neurogenesis
3. The MLP with periodic neurogenesis
4. The MLP with intermittently chaotic neurogenesis (α=3.82740)
5. The MLP with fully chaotic neurogenesis (α=4.0)

In the MLP with random neurogenesis, new neurons are introduced at random in the hidden layer. We use a Mean Square Error (MSE) as the measure of their performances. MSE is defined by Eq. (5).

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(t_n - o_n)^2. \quad (5)$$

We make a comparison between the performance of the conventional MLP and the proposed MLPs. Moreover, we carry out 100 trials with different initial weights of connections.

3.1. Learning Performance

In this section, we compare the learning performance of the five different MLPs. We show an example of the learning performance of the MLPs in Fig. 3. In this example, we set that the conventional MLP has 30 neurons in the hidden layer. Therefore, the proposed MLPs are set that the number of neurons in the hidden layer increases until 20 during the learning.

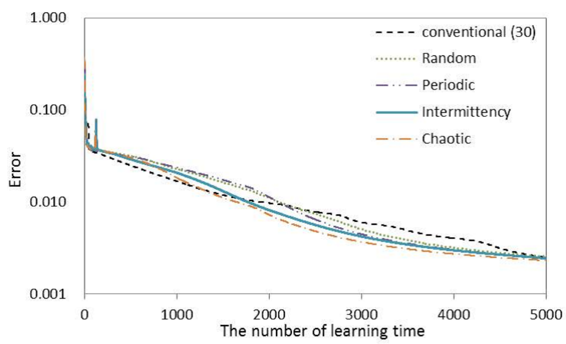

Figure 3: Learning performance of five MLPs.

From Fig. 3, the error of each MLP decreases. Moreover, we focus on the number of the neuron in the hidden layer. Therefore, we set that the conventional MLP is 30 neurons in the hidden layer. The proposed MLPs are set that the number of neurons in the hidden layer is 20 neurons at the stop of learning. However, we can see that the learning performance of the proposed MLPs are as well or better than the conventional MLP.

Therefore, we used the conventional MLP of 10, 20, 30, 40 and 50 neurons in the hidden layer. During the learning, the number of neurons in the hidden layer is increased until 20 and 30. We show the average of the learning performance of the MLPs in Table 1.

From Table 1 (1), we show the learning performance of the conventional MLP. We considered the best learning performance when we set 30 neurons in the hidden layer. Namely, we can say that 30 neurons is the best number of the neurons in the hidden layer. Therefore, we compare the conventional MLP and the proposed MLPs with 30 neurons in the hidden layer. From Table 1 (2), (3), (4) and (5), we show the learning performance of the proposed MLPs. We were able to obtain the performance of the proposed MLPs with 20 neurons as well or better than the conventional MLP with 30 neurons in the hidden layer. At the same time, we were able to obtain that the average of the processing time of the proposed MLPs are faster than the conventional MLP because used the small number of neurons in the hidden layer. Thus, we considered that the network of the quick processing time and high ability has been proposed. Because, we considered that the parameter of the connecting weight changed by the neurogenesis.

Table 1: Learning performance.
(a) The number of neurons in hidden layer. (b) Average of processing time. (c) Average of error. (d) Minimum error. (e) Maximum error. (f) St. Dev.

(1) The conventional MLP.

(a)	(b)	(c)	(d)	(e)	(f)
10	1.23	0.00813	0.00341	0.01460	0.00218
20	2.14	0.00321	0.00084	0.00747	0.00154
30	**3.04**	**0.00282**	0.00051	0.00809	**0.00161**
40	3.98	0.00296	0.00045	0.00784	0.00174
50	4.83	0.00330	0.00035	0.00910	0.00201

(2) The MLP with random neurogenesis.

(a)	(b)	(c)	(d)	(e)	(f)
20	**2.77**	**0.00307**	0.00085	0.00670	**0.00140**
30	3.65	0.00244	0.00055	0.00673	0.00142

(3) The MLP with periodic neurogenesis.

(a)	(b)	(c)	(d)	(e)	(f)
20	**2.07**	**0.00306**	0.00084	0.00948	**0.00160**
30	2.83	0.00248	0.00055	0.00662	0.00137

(4) The MLP with intermittently chaotic neurogenesis.

(a)	(b)	(c)	(d)	(e)	(f)
20	**2.13**	**0.00288**	0.00095	0.00818	**0.00148**
30	2.96	0.00244	0.00060	0.00792	0.00156

(5) The MLP with fully chaotic neurogenesis.

(a)	(b)	(c)	(d)	(e)	(f)
20	**2.13**	**0.00292**	0.00084	0.00670	**0.00121**
30	2.97	0.00234	0.00059	0.00792	0.00152

From Table 2, we focus on the conventional MLP with 30 neurons in the hidden layer. During the learning, the number of neurons in the hidden layer is increased until 20 at the stop of learning. We compare the learning performance of the proposed MLPs with the conventional MLP. We were able to obtain the performance of the proposed MLPs with 20 neurons as well or better than the conventional MLP with 30 neurons in the hidden layer. Thus, we were able to obtain the good performance which is the small number of neurons in the hidden layer.

Table 2: Average of pattern recognition performance.
(a) The number of neurons in hidden layer. (1) The conventional MLP. (2) The MLP with random neurogenesis. (3) The MLP with periodic neurogenesis. (4) The MLP with intermittency chaotic neurogenesis. (5) The MLP with fully chaotic neurogenesis.

(a)	Accuracy rate				
	(1)	(2)	(3)	(4)	(5)
20	95.15	94.70	95.44	97.66	97.07
30	95.68	98.05	98.20	97.66	98.49

3.2. Recognition Performance

In this section, we evaluate the recognition performance. Table 2 shows the average of pattern recognition of the 26 input patterns.

3.3. Parameter Dependency

In this section, we investigate the parameter dependency. We research change of some parameters by the proposed network.

3.3.1. Change of the parameter of the logistic map

We show the learning performance as change of α. α is the parameter of the logistic map. We show results of the change of α in Figs. 4 and 5.

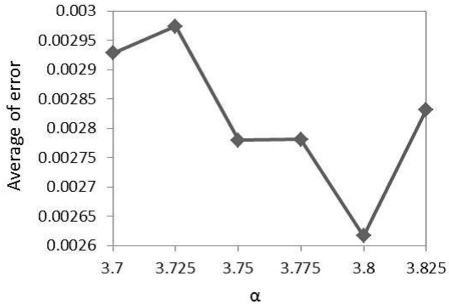

Figure 4: Learning performance as different α (between 3.7 and 3.825).

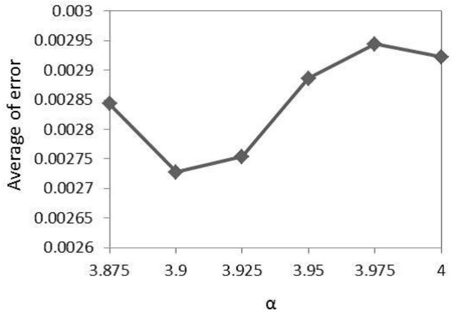

Figure 5: Learning performance as different α (between 3.875 and 4.0).

Figures 4 and 5 strike off the solution of the period 3 for the logistic map. From these figures, we can see that the error decreases in the vicinity of the solution of the period 3. However, we can not see the substantial change.

3.3.2. Change of the starting learning time of the neurogenesis

We show the learning performance as changing the start of learning time of the neurogenesis. When this parameter is small, the generation of neuron is quickly introduced.

From Fig. 6, we can see that the learning time of the quick start is better than the learning time of the slowly.

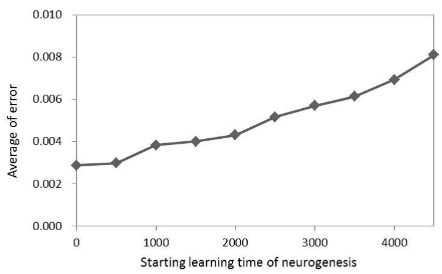

Figure 6: Learning performance as different the start learning iteration of the neurogenesis.

3.3.3. Change of the number of the neurogenesis by one generation

We show the learning performance as each number of the neurogenesis by one generation. When this parameter is large, more neurons are introduced in one generation.

Table 3: Learning performance as different the number of the neurogenesis by one generation.
(a) The number of neurons by one generation. (b) Average of error.

(a)	1	2	5	10
(b)	0.00288	0.00287	0.00257	0.00248

From Table 3, we can say that the error decreases by increasing the number of the generated neurons.

4. Conclusions

In this study, we showed the effectiveness of the proposed network with neurogenesis for pattern recognition. And we investigated the parameter dependency for detailed research on the influences of neurogenesis on MLP. We compared the five kinds of MLPs by pattern recognition.

We investigated in some approaches. Then, we were able to obtain the good performance by generating new neurons in the hidden layer during the learning process. However, we were able to obtain the learning performance also changed by the timing and number of neurogenesis.

Acknowledgment

This work was partly supported by MEXT/JSPS Grant-in-Aid for JSPS Fellows (24·10018).

References

[1] S. Becker, J. M. Wojtowicz, "A Model of Hippocampal Neurogenesis in Memory and Mood Disorders," Cognitive Sciences, vol. 11, no. 2, pp. 70-76, 2007.

[2] R. A. Chambers, M. N. Potenza, R. E. Hoffman, W. Miranker, "Simulated Apotosis/Neurogenesis Regulates Learning and Memory Capabilities of Adaptive Neural Networks," Neuropsychopharmacology, pp. 747-758, 2004.

[3] H. Satoi, H. Tomimoto, R. Ohtani, 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.

[4] Y. Yokoyama, T. Shima, C. Ikuta, Y. Uwate and Y. Nishio, "Improvement of Learning Performance of Neural Network Using Neurogenesis," Proceedings of RISP International Workshop on Nonlinear Circuits and Signal Processing (NCSP'12), pp. 365-368, Mar. 2012.

[5] Y. Yokoyama, C. Ikuta, Y. Uwate and Y. Nishio, "Performance of Multi-Layer Perceptron with Neurogenesis," Proceedings of International Symposium on Nonlinear Theory and its Applications (NOLTA'12), pp.715 718, Oct. 2012.

[6] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Representations by Back Propagation Error," Nature, vol. 323-9, pp. 533-536, 1986.

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

[8] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, "Parallel distributed processing," MIT Press, 1986.

[9] M Hasegawa, T Ikeguchi, T Matozaki and K Aihara, "An analysis on additive effects of nonlinear dynamics for combinatorial optimization," IEICE Trans. Fundamentals, vol. E80-A, no. 1, pp. 206-213, 1997.