

IEICE Proceeding Series

Performance of Multi-Layer Perceptron with Neurogenesis

Yuta Yokoyama, Chihiro Ikuta, Yoko Uwate, Yoshifumi Nishio

Vol. 1 pp. 715-718

Publication Date: 2014/03/17

Online ISSN: 2188-5079

Downloaded from www.proceeding.ieice.org

©The Institute of Electronics, Information and Communication Engineers

Performance of Multi-Layer Perceptron with Neurogenesis

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. The new neurons create new network. It is known that the neurogenesis causes the improvement of memory, learning, and thinking ability by combining new neurons with biological neural network. We consider that the neurogenesis can be applied to an artificial neural network.

In this study, we propose the Multi-Layer Perceptron (MLP) with neurogenesis and apply to pattern recognition. In the MLP with neurogenesis, some neurons are generated in a hidden layer. We propose random, periodic and chaotic timing methods to introduce neurogenesis. We compare the performance of the MLP with neurogenesis with the conventional MLP.

1. Introduction

In the human brain, neurons had been considered to be lost with age until several years ago. However, in recent studies, some researchers reported that new neurons are generated in the dentate gyrus of hippocampus [1] [2]. This process is called “neurogenesis”. It is reported that this process occur all human brains. By utilizing the neurogenesis, some brain cells increase and the network of within is substantial. It is known that the neurogenesis improves ability to solve problems like memory and thinking ability by connection of new neurons.

In this study, we apply the behavior of neurogenesis to the Multi-Layer Perceptron (MLP) which is a famous feed-forward neural network. In the proposed neural network, some new neurons are generated in a hidden layer. We name this network “MLP with neurogenesis.” In order to confirm the efficiency of neurogenesis, we investigate the performance of MLP with neurogenesis for learning several alphabet patterns. We confirm that the MLP with neurogenesis obtains better results than the conventional MLP.

2. MLP with Neurogenesis

A Multi-Layer Perceptron (MLP) is one of a feed-forward neural network. MLP is a most famous feed-forward neural network. This network is used for pattern recognition, pattern classification, pattern learning, and other tasks. MLP is composed some layers, 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 [3] [4]. In this algorithm, the network calculate the error from the output and teaching signal. After that, this error is propagating backward in the network. The network can learn to tasks by the repeating this process.

In this study, we consider which composed of three layers (one input, one hidden, and one output layer) MLP. In the MLP with neurogenesis, some neurons are generated in a hidden layer. Figure 1 shows a structure of the MLP with neurogenesis.

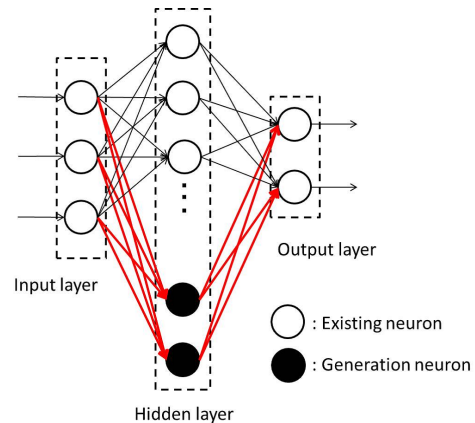


Figure 1: MLP with neurogenesis.

2.1. Updating Rule of Neuron

The updating rule of neuron is described by Eq. (1).

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

where x is the input or output and w is the connection weight parameter and θ is threshold. The sigmoid function is described by Eq. (2). This is used for the output function.

$$f(a) = \frac{1}{1 + e^{-a}}. \quad (2)$$

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 Eqs. (3) and (4).

$$E = \sum_{p=1}^p E_p, \quad (3)$$

$$E_p = \frac{1}{2} \sum_{i=1}^n (t_{pi} - o_{pi})^2, \quad (4)$$

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 p th input data, and o_{pi} is the value of the output data for the p th input data. The connection weight is described by Eq. (5).

$$\Delta_p w_{ij}^{k-1,k} = \eta_{pj}^k o_{pi}^{k-1} = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}}, \quad (5)$$

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

2.2. Neurogenesis

The neurogenesis had been considered to generate for period of growth. However, the neurogenesis in the hippocampus of the human brain was discovered in the late 1990s by Erickson et al [1] [2]. Before that time, the neuron had been considered to be lost with age. The neurogenesis is that new neurons are generated in the human brain. The neurogenesis causes the improve memory, learning, thinking ability, and so on. We assumed that the MLP can be effective performance by introducing the neurogenesis.

In this study, we apply the behavior of neurogenesis to the MLP. We explain how to introduce neurogenesis. In this system, new neurons are generated in the hidden layer. At the same time, all the weights connecting to the generated neurons are newly set between -1.0 and 1.0 at random. In this study, the process to generate neurons and connection is ‘‘neurogenesis.’’ After that, the connection weights are newly calculated.

We explain the learning methods of periodic and chaotic neurogenesis. The periodic neurogenesis generates the new neuron at every 50 iteration during the learning process. In the case of chaotic neurogenesis, the new neuron is generated by using the logistic map. The updating function of the logistic map is described by Eq. (6).

$$y(n+1) = \alpha y(n)(1-y(n)), \quad (6)$$

In this study, we use that the parameter $\alpha=3.8250$ and 4.0. When we choose that the parameter α is 3.8250, it is well known that the map produces intermittent bursts just before

periodic-windows. Figures 2 and 3 show chaotic time series for $\alpha=3.8250$ and 4.0. In this timing method, the neurogenesis is occurred when the value of y takes the range between 0.6 to 0.61 of Figs. 2 and 3.

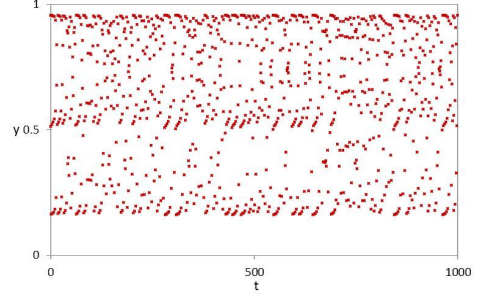


Figure 2: Chaotic time series ($\alpha=3.8250$).

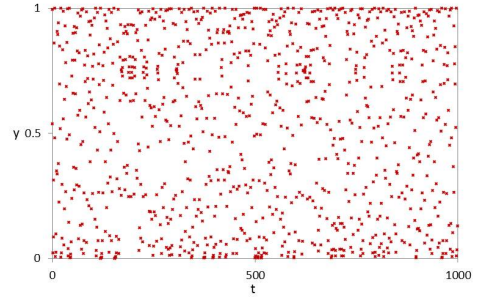


Figure 3: Chaotic time series ($\alpha=4.0$).

3. Simulations

In this study, we consider pattern recognition. Our BP learning was based on the following parameters. We consider that we propose network is composed of three layers. The number of neurons in the input layer and output layer are 35 and 26. Similarly, we set 20 neurons in the hidden layer at the start of learning. The maximum number of neurons in the hidden layer is set to 50. The learning time is set to $m = 10000$. The learning rate is $\eta = 0.005$, respectively, and initial value of the weight are given between -1.0 and 1.0 at random. Moreover, we compare the learning performance of five kinds MLPs:

- (1) The conventional MLP
- (2) The MLP with random neurogenesis
- (3) The MLP with periodic neurogenesis
- (4) The MLP with chaotic neurogenesis ($\alpha=3.8250$)
- (5) The MLP with chaotic neurogenesis ($\alpha=4.0$)

In the MLP with random neurogenesis, this MLP is introduced the new neurons at random until 50 neurons in the hidden layer. Namely, the proposed MLPs are set the number of maximum neurons in the hidden layer to 50 at the

end of learning. I use a Mean Square Error (MSE) to the measures of performance. MSE is described by Eq. (7).

$$MSE = \frac{1}{N} \sum_{n=1}^N (t_n - o_n)^2, \quad (7)$$

We make a comparison between the performance of the conventional MLP and the proposed the MLPs. Figure 4 shows input patterns. In this study, we prepare 26 learning patterns. We used the input patterns of alphabet A to Z. Moreover, we simulate the 100 trials from different initial weights of connection.

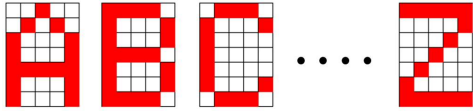


Figure 4: Input patterns.

3.1. Learning Performance

We compare the five different MLPs. Moreover, we used 10, 20, 30, 40, 50 and 60 neurons in the hidden layer for comparison. We show the learning performance of MLPs in Table 1. From Table 1, we can see that the learning

Table 1: Learning performance. (a) The number of neurons in hidden layer. (b) Average of error. (c) Minimum error. (d) Maximum error.

	(a)	(b)	(c)	(d)
(1)	10	0.0798	0.0755	0.0836
	20	0.0777	0.0703	0.0836
	30	0.0734	0.0664	0.0796
	40	0.0697	0.0630	0.0785
	50	0.0650	0.0587	0.1039
(2)	20→50	0.0601	0.0542	0.0689
(3)	20→50	0.0223	0.0177	0.0276
(4)	20→50	0.0262	0.0175	0.0585
(5)	20→50	0.0222	0.0175	0.0286
(5)	20→50	0.0217	0.0179	0.0285

performance of the proposed MLP is the best of all. The learning performance of the conventional MLP is the worst. However, we were able to obtain the best learning performance by the MLP with chaotic neurogenesis ($\alpha=4.0$). We consider that the good results were obtained by generating the new neurons and irregular timing. Moreover, we compare the conventional MLP set 50 neurons in the hidden layer with the proposed MLPs. The results of the proposed MLPs are good although the number of neurons is equal after learning. We were able to obtain the good performance by generated the new neurons during the learning.

3.2. Pattern Recognition

In this section, we show the pattern recognition. Table 2 shows the performance of pattern recognition to each input patterns.

Table 2: Recognition performance. (a) The number of neurons in hidden layer.

		Accuracy rate [%]					
		Pattern					
	(a)	A	B	C	D	E	
(1)	20	89.58	94.81	93.31	93.15	94.81	
	50	87.27	92.15	89.92	90.27	91.42	
(2)	20→50	84.62	92.31	92.31	80.77	88.46	
(3)	20→50	88.46	80.77	84.62	73.08	80.77	
(4)	20→50	88.46	96.15	88.46	92.31	96.15	
(5)	20→50	100.0	96.15	88.46	100.0	96.15	
		Accuracy rate [%]					
		Pattern					
	(a)	F	G	H	I	J	
(1)	20	92.62	92.65	92.58	88.00	88.19	
	50	90.38	89.58	90.27	87.42	87.58	
(2)	20→50	84.62	92.31	88.46	88.46	92.31	
(3)	20→50	80.77	88.46	80.77	84.62	88.46	
(4)	20→50	84.62	92.31	92.31	92.31	92.31	
(5)	20→50	96.15	92.31	96.15	96.15	96.15	
		Accuracy rate [%]					
		Pattern					
	(a)	K	L	M	N	O	
(1)	20	90.00	92.65	93.19	93.35	93.88	
	50	88.38	88.46	90.27	91.08	90.04	
(2)	20→50	84.62	84.62	88.46	96.15	84.62	
(3)	20→50	80.77	84.62	76.92	76.92	80.77	
(4)	20→50	92.31	88.46	88.46	92.31	92.31	
(5)	20→50	100.0	96.15	96.15	100.0	96.15	
		Accuracy rate [%]					
		Pattern					
	(a)	P	Q	R	S	T	
(1)	20	92.77	92.04	94.19	92.92	88.00	
	50	90.38	89.50	89.65	89.50	87.81	
(2)	20→50	88.46	88.46	84.62	96.15	96.15	
(3)	20→50	76.92	80.77	76.92	84.62	80.77	
(4)	20→50	88.46	88.46	88.46	92.31	76.92	
(5)	20→50	96.15	96.15	96.15	100.0	100.0	
		Accuracy rate [%]					
		Pattern					
	(a)	U	V	W	X	Y	Z
(1)	20	93.85	93.31	93.54	84.73	86.12	91.42
	50	90.65	89.69	90.46	86.38	87.27	91.31
(2)	20→50	92.31	84.62	88.46	88.46	96.15	88.46
(3)	20→50	84.62	84.62	88.46	76.92	80.77	80.77
(4)	20→50	88.46	80.77	88.46	80.77	92.31	96.15
(5)	20→50	96.15	96.15	96.15	100.0	100.0	96.15

Table 3: Average of recognition performance. (a) The number of neurons in hidden layer. (b) Average. (c) Minimum. (d) Maximum.

		Accuracy rate [%]			
		(a)	(b)	(c)	(d)
(1)	20	91.76	84.73	94.81	
	50	89.50	87.27	92.15	
(2)	20→50	89.05	80.77	96.15	
(3)	20→50	81.80	73.08	88.46	
(4)	20→50	89.64	76.92	96.15	
(5)	20→50	96.89	88.46	100.0	

From Table 2, we can say that the recognition performance of the proposed MLPs are better than the conventional MLP in a certain case. Especially, we were able to obtain 100% by the proposed MLP. And, we show the average, value of minimum, and maximum in Tab 3 which is

calculated from Tab. 2.

From Table 3, we can say that the proposed MLPs are similar to the conventional MLP. We consider that the learning is mostly possible of all MLPs.

3.3. Generalization Capability

In this section, in order to evaluate the generalization capability, we prepare the input pattern which gave noise of 7 bits to each input patterns. Each result shows an average of 100 trials. Table 4 shows the generalization capability of pattern recognition to each input patterns.

Table 4: Generalization capability.
(a) The number of neurons in hidden layer.

		Accuracy rate [%]					
		Pattern					
	(a)	A	B	C	D	E	
(1)	20	88.46	92.31	92.31	96.15	92.31	
	50	80.77	92.31	96.15	84.62	84.62	
(2)	20→50	92.31	80.77	76.92	84.62	88.46	
(3)	20→50	81.38	80.15	83.00	84.00	82.38	
(4)	20→50	88.46	88.46	96.15	92.31	84.62	
(5)	20→50	88.46	92.31	92.31	92.31	88.46	
	(a)	F	G	H	I	J	
(1)	20	92.31	96.15	88.46	76.92	88.46	
	50	88.46	96.15	88.46	80.77	88.46	
(2)	20→50	88.46	76.92	84.62	76.92	88.46	
(3)	20→50	75.31	84.62	80.15	80.77	88.46	
(4)	20→50	92.31	96.15	84.62	88.46	88.46	
(5)	20→50	84.62	92.31	88.46	84.62	80.77	
	(a)	K	L	M	N	O	
(1)	20	96.15	92.31	76.92	84.62	96.15	
	50	76.92	84.62	96.15	92.31	96.15	
(2)	20→50	96.15	88.46	88.46	96.15	84.62	
(3)	20→50	82.38	84.00	76.92	73.08	84.00	
(4)	20→50	84.62	96.15	88.46	88.46	96.15	
(5)	20→50	84.62	84.62	92.31	92.31	92.31	
	(a)	P	Q	R	S	T	
(1)	20	84.62	96.15	76.92	92.31	73.08	
	50	88.46	84.62	88.46	96.15	84.62	
(2)	20→50	88.46	92.31	88.46	84.62	73.08	
(3)	20→50	72.46	81.38	70.23	81.77	75.31	
(4)	20→50	92.31	92.31	92.31	96.15	96.15	
(5)	20→50	88.46	88.46	92.31	84.62	80.77	
	(a)	U	V	W	X	Y	Z
(1)	20	96.15	92.31	92.31	73.08	73.08	80.77
	50	84.62	92.31	92.31	80.77	84.62	88.46
(2)	20→50	96.15	92.31	88.46	84.62	92.31	76.92
(3)	20→50	84.00	84.00	80.77	71.46	77.54	80.77
(4)	20→50	88.46	88.46	88.46	92.31	92.31	92.31
(5)	20→50	92.31	92.31	92.31	84.62	88.46	88.46

From Table 4, we can say that the proposed MLPs are comparatively better than the conventional MLP. Moreover, we show the average, value of minimum, and maximum in Tab 5 which is calculated from Tab. 4.

From Table 5, we can say that the MLP with chaotic neurogenesis is the best as average of accuracy rate. Moreover,

Table 5: Average of generalization capability.
(a) The number of neurons in hidden layer. (b) Average. (c) Minimum. (d) Maximum.

		Accuracy rate [%]			
		(a)	(b)	(c)	(d)
(1)	20	87.72	73.08	96.15	
	50	88.17	76.92	96.15	
(2)	20→50	86.54	73.08	96.15	
(3)	20→50	80.01	70.23	88.46	
(4)	20→50	90.98	84.62	96.15	
(5)	20→50	88.61	88.46	92.30	

the average of the MLPs with chaotic neurogenesis is better than the conventional MLP. However, MLP with chaotic neurogenesis ($\alpha=4.0$) network's maximum accuracy rate is worst.

4. Conclusions

In this study, we applied the behavior of neurogenesis to the MLP which is a famous feed-forward neural network. In the proposed neural network, some new neurons are generated in a hidden layer by effect neurogenesis. We proposed random, periodic and chaotic timing methods to introduce neurogenesis. In order to confirm the efficiency of neurogenesis, we investigated the performance of MLP with neurogenesis for learning several alphabet patterns.

By computer simulations, we showed improvement of learning performance by the proposed MLPs. In the pattern recognition, we were able to obtain the results almost equivalent to the conventional MLP by proposed MLPs. Moreover, we confirmed that the proposed MLPs with neurogenesis obtained better learning performance and generalization capability than the conventional MLP. Thus, we consider that the neurogenesis can gave good influence to the MLP learning. As a future work, we would like to clarify between chaotic and random neurogenesis.

Acknowledgments

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 Apoptosis/Neurogenesis Regulates Learning and Memory Capabilities of Adaptive Neural Networks," *Neuropsychopharmacology*, pp. 747-758, 2004.
- [3] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Representations by Back Propagation Error," *Nature*, vol. 323-9, pp. 533-536, 1986.
- [4] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, "Parallel distributed processing," MIT Press, 1986.