Journal of Electrical Engineering 6 (2018) 289-294
doi: 10.17265/2328-2223/2018.05.006

Learning Performance of Linear and Exponential Activity Function with Multi-layered Neural Networks

Betere Job Isaac[1], Hiroshi Kinjo[2], Kunihiko Nakazono[2] and Naoki Oshiro[2]

1. Mechanical Systems Engineering Course, Graduate School of Engineering and Science, University of the Ryukyus Senbaru 1, Nishihara, Okinawa 903-0213, Japan.

2. Faculty of Engineering, University of the Ryukyus Senbaru 1, Nishihara, Okinawa 903-0213, Japan

Abstract: This paper presents a study on the improvement of MLNNs (multi-layer neural networks) performance by an activity function for multi logic training patterns. Our model network has L hidden layers of two inputs and three, four to six output training using BP (backpropagation) neural network. We used logic functions of XOR (exclusive OR), OR, AND, NAND (not AND), NXOR (not exclusive OR) and NOR (not OR) as the multi logic teacher signals to evaluate the training performance of MLNNs by an activity function for information and data enlargement in signal processing (synaptic divergence state). We specifically used four activity functions from which we modified one and called it L & exp. function as it could give the highest training abilities compared to the original activity functions of Sigmoid, ReLU and Step during simulation and training in the network. And finally, we propose L & exp. function as being good for MLNNs and it may be applicable for signal processing of data and information enlargement because of its performance training characteristics with multiple training logic patterns hence can be adopted in machine deep learning.

Key words: Multi-layer neural networks, learning performance, multi logic training patterns, Activity function, BP neural network, deep learning.

1. Introduction

Neural networks have been proved by an increasing number of researchers [1-3] especially for signal processing as to manage data transfer effectively with other systems to overcome conventional means particularly [4]. Many scientists have undergone a resurgence in computation research community from the field of neural networks and machine learning. Furthermore, recently more research has been reported on multi hidden layer neural network [5-7]. Earlier scientists were motivated in large part by visions of imbuing computer programs with life-like ability to self-replicate and with adaptive capability to learn the environment [8-10]. And results showed some degradation in the results of the sigmoid function. It is also said that ReLU function is good compared to other activity functions in 2D image processing. It is also noted that Step function is not tried due to the fact of no derivative characteristics for BP training. Therefore, our motivation interest was to investigate the performance of an Activity function using MLNNs (multi-layer neural networks) and confirm with regards to the findings. In this study, we propose an activity function to overcome the drawbacks like gradient disappearance problem of sigmoid function and the weakness of ReLU functions [11] like being limited to some training patterns with this type of neural network training structure.

It is said that large convolution structure is very popular now by ReLU activity function but our motivation in this study is to really find out the activity function training with MLNNs without depending on data set neural networks training with an aim of improving the activity function training performance.

Corresponding author: Betere Job Isaac, Ms., research fields: robotics, artificial intelligent control systems and signal processing.

2. Neural Network Model

Neural networks are typically organized in layers.

Fig. 1 shows the MLNN model we used in this study. Layers are made up of several interconnected nodes which contain an activation function. Patterns are presented to the network via the input layer which communicates to 5 or more neurons and 5 hidden layers where the actual processing is done via a system of weighted connections respectively. The hidden layers then link to an output layer and give the result of the desired output as shown in the multi-layered feedforward NN in Fig. 1 where I, J & K are the number of neurons of input layer, hidden layers and output layer respectively with L hidden layers. The logic functions of XOR, OR, AND, NAND, NXOR & NOR are used as teacher training signals.

The input/output relation of the NN is given by the following equations.

$$o_i = f^I(x_i),\ i = 1, 2, \dots, I \tag{1}$$

$$o_j{}^{(l)} = f^H\left(net_j{}^{(l)}\right), l = 1,2, \dots, L, j = 1, 2, \dots, J \tag{2}$$

$$net_j{}^{(l)} = \begin{cases} \sum_i w_{ji} o_i, & l = 1 \\ \sum_j w_{jj}{}^{(l)}\, o_j{}^{(l-1)}, & l \geq 2 \end{cases}$$

$$o_k = f^O(net_k), k = 1, 2, \dots, K \tag{3}$$

$$net_k = \sum_j w_{kj}\, o_j{}^{(L)}$$

where o_i and o_j are outputs of input and hidden layers respectively, w_{ji} w_{jj} and w_{kj} are connecting weights. $f^I(.), f^H(.)$ and $f^O(.)$ are activity functions of input, hidden and output layers respectively.

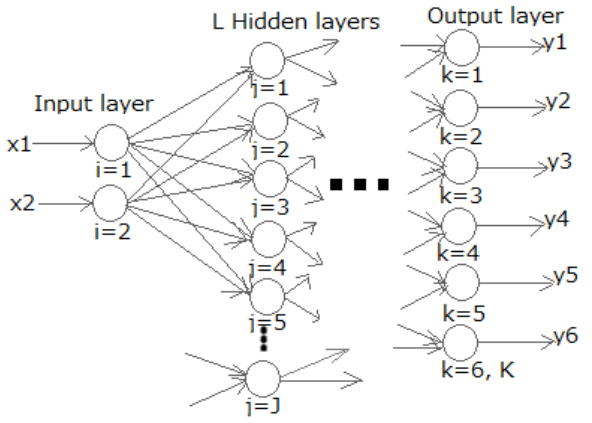

Fig. 1 Multi-layered neural network.

Many methods of bio-inspired neural networks for signal processing have been well studied and applied for many industrial problems. There are many network types consisting of many inputs with few outputs and it is useful for image processing. However, we consider the other type of the network construction as in Fig. 1. The network has a larger number of outputs than inputs and does not depend on data sets neural network structure. This network may be applicable to data enlargement fields. Simulations were basically concentrated on basic activity functions as follows.

Sigmoid function:

$$f^H(x) = \frac{1}{1+exp^{(-x)}} \tag{4}$$

ReLU function:

$$f^H(x) = \begin{cases} x, & 0 \leq x \\ 0, & 0 > x \end{cases} \tag{5}$$

Step function:

$$f^H(x) = \begin{cases} 1, & 0 \leq x \\ 0, & 0 > x \end{cases} \tag{6}$$

And the L & exp function as

$$f(x) = \begin{cases} x + \beta, & x \geq 0 \\ \beta\, e^x, & x < 0 \end{cases} \tag{7}$$

where β is the intercept. This function combined both linear part and exponential part, so we called it L & exp function.

In this study, we have used the input and output activity function as follows:

$$f^I(x) = f^O(x) = x \tag{8}$$

3. BP (Backpropagation) for MLNNs

BP training is a gradient descent algorithm. It tries to improve the performance of the neural net by reducing its error along its gradient. The error is expressed by the RMS (root-mean-square) error, which can be calculated by the error function E for BP as shown.

$$\boldsymbol{E} = \frac{1}{2} \sum_{p=1}^{p} \sum_{k=1}^{K} \left(t_k{}^{(p)} - o_k\right)^2 \tag{9}$$

where the error E is half the sum of the geometric averages of the difference between the desired output

$t_k^{(p)}$ and the actual output o_k over all patterns p. In each training step, the weights w_{ji}, w_{jj} and w_{kj} are adjusted towards the direction of maximum decrease and scaled by some learning rate epsilon ε as shown in the following modified equation of the synaptic connection weight vector W.

$$W^{(new)} = W^{(old)} - \varepsilon \frac{\partial E}{\partial W} \quad (10)$$

The generalized delta rule of BP is applied, and the gradient is as follows with the output layer:

$$\frac{\partial E}{\partial W} = -\delta_k \, o_j^{(l)} \quad (11)$$

where δ_k is as follows

$$\delta_k = (t_k - o_k) f^{o\prime}(net_k) \quad (12)$$

and, for the hidden layers as follows:

$$\frac{\partial E}{\partial W} = -\delta_j^{(l)} o_j^{(l-1)} \quad (13)$$

where $\delta_j^{(l)}$ is as follows

$$\delta_j^{(l)} = \sum \{\delta_j^{(l+1)} w_{jj}^{(l+1)}\} f^{H\prime}\left(net_j^{(l)}\right), l = L-1, L-2, \ldots, 2, 1 \quad (14)$$

where $f^{H\prime}(.)$ is used in the following derivative functions:

Derived sigmoid function:

$$f^{H\prime}(x) = f^H(x)(1 - f^H(x)) \quad (15)$$

Derived ReLU function as:

$$f^{H\prime}(x) = \begin{cases} 1, & 0 \le x \\ 0, & 0 > x \end{cases} \quad (16)$$

For the step function as Eq. (6), we assumed the following derived Step function:

$$f^{H\prime}(x) = \begin{cases} 1, & 0 \le x \\ 0, & 0 > x \end{cases} \quad (17)$$

By assuming the derivative function of step function, it enables BP training for Step function as an activity function in neural networks.

And the derived L & exp function as

$$f^{H\prime}(x) = \begin{cases} 1, & x \ge 0 \\ \beta\, e^x, & x < 0 \end{cases} \quad (18)$$

4. Experiment Simulations

We considered the training of basic logic functions to be the fundamental task to discuss the performance of neural networks without depending on data sets neural network structure such that our results can be easily confirmed.

Tables 1-3 show the parameters, Tables 4-6 show the training patterns as inputs and outputs during training in the network with t_1, t_2, t_3 and t_4 as teacher signals for three to four outputs when parameter J = 5 and with t_1, t_2, t_3, t_4, t_5 and t_6 as teacher signals for six outputs when parameter J = 12. Results were obtained as reflected in Tables 7-9 showing the successive rate percentages where L is the number of hidden layers. Figs. 2-4 show the training results where E is the error function. The successive stop condition is when $E < 0.001$. It is seen that L & exp. is better than other basic activity functions for the basic multi logic training pattern outputs in performance with the MLNNs. It is also noted that Step function could also train with BP when its original function is taken to be its derivative. It is also seen that Step function could train with the patterns in the 1st layer for all outputs with quiet good results but degrades highly when layers and neurons increase. This would be an advantage because it has always been that step function does not train with BP by many scientists. ReLU function could not train patterns with four and six outputs indicating a high degradation training results and worst performance in the network.

Table 1 Constant parameter of the MLNNs.

No.	Parameters	Value/method
1	No. of neurons in input layer (I)	2
2	No. of neurons in hidden layer (J)	5 or 12
3	No. of hidden neuron layers (L)	1-5
4	No. of neurons in output layer (K)	3, 4 or 6
5	Activity functions	Sigmoid ReLU Step L & exp (β = 0.2)

Table 2 Constant parameter of the BP.

No.	Parameters	Value/method
1	Training coefficient (ε)	0.1
2	Iterations	3,000

Table 3 Training parameters.

No.	Teacher signals					
1	t_1	t_2	t_3	t_4	t_5	t_6
2	XOR	AND	OR	NAND	NXOR	NOR

Table 4 Training pattern for three outputs.

No.	Inputs		Teacher signals		
	X_1	X_2	t_1	t_2	t_3
1	0	0	0	0	0
2	0	1	1	0	1
3	1	0	1	0	1
4	1	1	0	1	1

Table 5 Training patterns for four outputs.

No.	Inputs		Teacher signals			
	X_1	X_2	t_1	t_2	t_3	t_4
1	0	0	0	0	0	1
2	0	1	1	0	1	1
3	1	0	1	0	1	1
4	1	1	0	1	1	0

Table 6 Training patterns for six outputs.

No.	Inputs		Teacher signals					
	x_1	x_2	t_1	t_2	t_3	t_4	t_5	t_6
1	0	0	0	0	0	1	1	1
2	0	1	1	0	1	1	0	0
3	1	0	1	0	1	1	0	0
4	1	1	0	1	1	0	1	0

Table 7 Successive rate for three outputs [%].

L	1	2	3	4	5
Sigmoid	100	74	0	0	0
ReLU	65	46	38	21	13
Step	74	4	3	1	0
L&Exp.	100	95	95	91	72

Table 8 Successive rate for four outputs [%].

L	1	2	3	4	5
Sigmoid	100	69	0	0	0
ReLU	0	0	0	0	0
Step	77	11	0	0	0
L&Exp.	100	100	99	88	43

Table 9 Successive rate for six outputs [%].

L	1	2	3	4	5	6
Sigmoid	100	100	11	0	0	0
ReLU	0	0	0	0	0	0
Step	99	12	2	0	0	0
L&Exp.	100	100	99	100	100	92

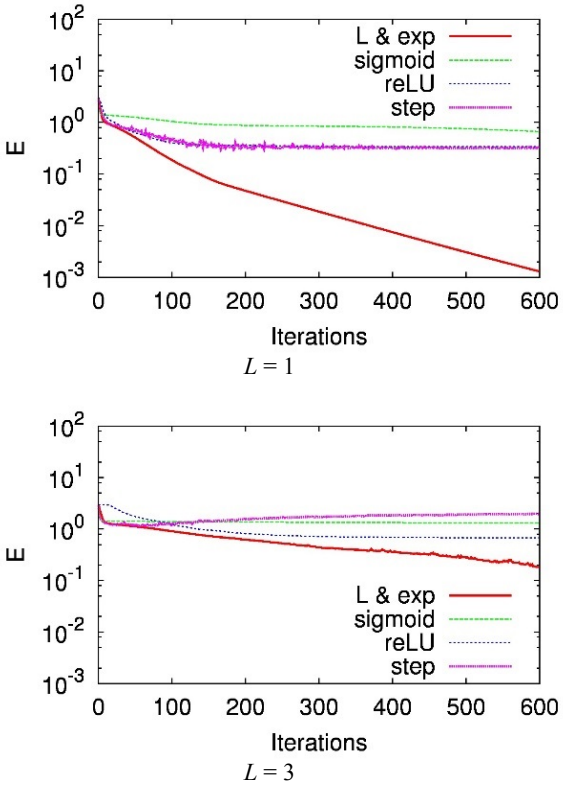

Fig. 2 Training results for three outputs.

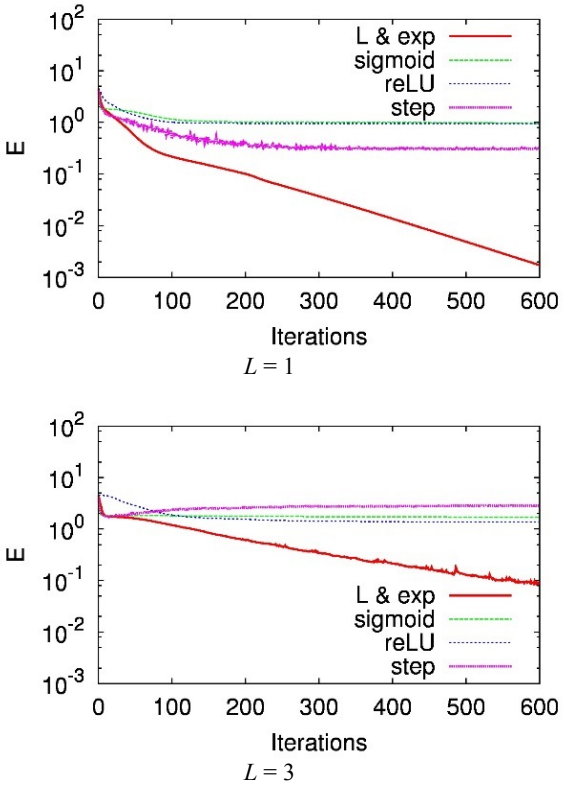

Fig. 3 Training results for four outputs.

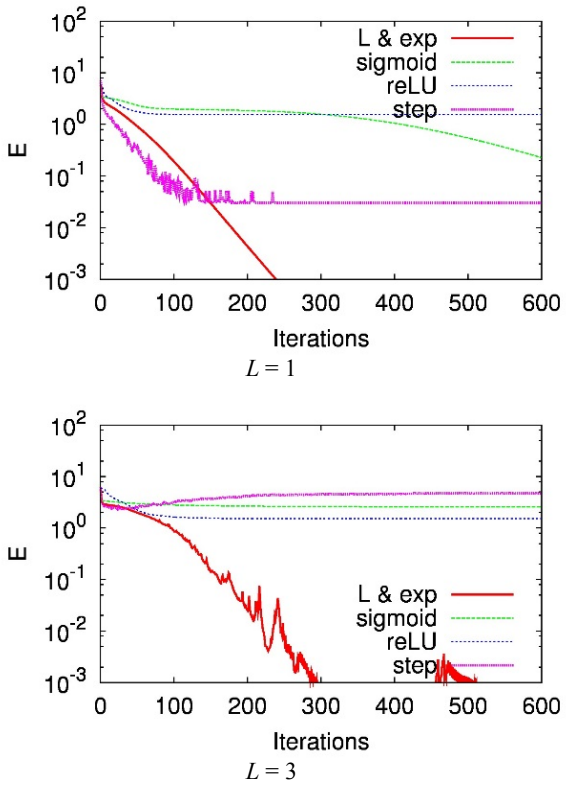

Fig. 4 Training results for six outputs.

5. Discussion

As per the results after training, it is seen in Tables 7-9 that L & exp function trains all the patterns for the basic multi logic training patterns with the highest percentage learning rate in the network. Sigmoid could only train with two layers only in the network because of the gradient disappearance problem. ReLU function trained with three-output case only but could not train with a four- and six- multi logic training pattern case as seen in Tables 8 and 9. The successive count training is worse and degrades highly with Sigmoid, ReLU and Step due to the fact that some nonlinear functions are limited to some training patterns and cannot give the desired output in such network which is seen as a negative effect in image and signal processing. Step function is seen to have the highest number of degradations in our network model but was good with few layers in all outputs hence being with a positive effect and has an advantage of low computational cost and easy implementation in computer hardware.

We can see that Sigmoid, ReLU and Step functions could not train satisfactorily as compared to the proposed L & exp. function. We considered the fact that L & exp function could successfully train because it has no limitation to the gradient values whereby it accommodates both positive and negative values with the increase in the number of layers in the network when applied to different tasks with various patterns. We have also analyzed all patterns using Eqs. (1)–(3) and seen that it cannot give the desired output with some training patterns and gives a fetal error in training especially with ReLU activity function.

6. Conclusion

In this study, we investigated the multi-layered neural network learning performance by an activity function and solved the drawback of some basic nonlinear activity functions. It is noted that the BP training gives better training results in signal processing by an activity function with few inputs to the basic multi logic outputs as proved by this study for either three outputs, four outputs or six outputs training network using L & exp, Sigmoid, ReLU and Step as activity functions. This showed that L & exp activity function network trained all the patterns for all outputs in MLNNs without any interference as compared to the rest of the basic activity functions used in this study hence proposed to handle large volumes of parameters in machine deep learning. However, in this study, we have seen that there are some outputs that could not be trained due to the weakness and disadvantages of ReLU function being limited to some patterns. For Step function, we assumed a derivative function as expressed by Eq. (17) and showed error cumulations with BP training network and resulted to fading and degradation of training performance with MLNNs when the numbers of layers and neurons are increased especially for four outputs and six outputs training networks. The worst learning performance of these activity functions with some training patterns is noted as a fetal error which requires a mathematical

analytical method investigation. As for the future work, we shall apply the proposed L&exp. function to convolution neural network structure as the activity function and investigate the performance so that we can integrate and develop other artificial intelligent systems in deep learning.

References

[1] Rumelhart, D. E., McClelland, J. L., and the PDP Research Group. 1986. *Parallel Distribution Processing*, MIT Press.

[2] Hassoun, M. H. 1995. *Fundamentals of Artificial Neural Networks*, MIT Press.

[3] Anderson, J. A., and Rosenfeld, E. 1988. *Neuro Computing Foundations of Research*, MIT Press.

[4] Lewis, F. L., Jagannathan, S., and Yesidirek, A. 1999. *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, 1798-998.

[5] Kodaka, T., and Murakami, K. 2016. *Machine Learning and Deep Learning, Simulation by C Programming*. Ohmsha. (in Japanese)

[6] Okatani, T. 2015. *Deep Learning*, Kodansha. (in Japanese)

[7] Kamishima, T., Asoh, H., Yasuda, M., Maeda, S., Okanohara, D., Okatani, T., Kubo, Y., and Bollegala, D. 2015. *Deep Learning.* (in Japanese)

[8] Albrecht, R. F., Reeves, C. R., and Steele, N. C., eds. 1993. *Artificial Neural Nets and Genetic Algorithms*. Adolf Holzhausens Nachfolger, A-1070 Wien, Austria.

[9] Lin, C. T., and Lee, C. S. G. 1996. *Neural Fuzzy Systems, a Neural-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall, Inc. A Simon & Shuster Company Upper Saddle River, NJ07458, USA.

[10] Asakawa, S. 2016. *Practical Python Recipes of Deep Learning*. Tokyo: Corona Publishing Co. Ltd.

[11] Betere, J, I., Kinjo, H., Nakazono, K., et al. 2018. *Investigation of Multi-Layers Neural Network Performance Evolved by Genetic Algorithms*. Artif Life Robotics. https://doi.org/10.1007/s10015-018-0494-2, Japan.