2017 RISP International Workshop on Nonlinear Circuits,
Communications and Signal Processing (NCSP'17)
Guam, USA, February 28th to March 3rd, 2017

Improvement of Learning Efficiency Using Noise for Back Propagation in Deep Learning

Kazuki NAGAO, Shinsaburo KITTAKA, Yoko UWATE and Yoshifumi NISHIO

Department of Electrical and Electronic Engineering of Tokushima University
2-1 Minami-Josanjima, Tokushima, Tokushima 770-8506, Japan
E-mail: {k-nagao, kittaka, uwate, nishio}@ee.tokushima-u.ac.jp

Abstract

Deep learning is made to imitate the human nervous system. Since 1980s, there have been concepts and methods of deep learning. Currently, deep learning is becoming the third artificial intelligence boom. In this study, we propose a system of deep learning. This system is noisy back propagation. Weights of neurons are optimized using a noise in the back propagation. The purposes of this study are to improve learning accuracy and to reduce learning loops. We use two data set. We compare the proposed system with the conventional system.

1. Introduction

Human nervous system has neurons which are connected. Then, human can think things. Deep learning is made to imitate the human nervous system. Since 1980s, there have been concepts and methods of deep learning. Deep learning was paid a lot of attention in the field of image recognition from 2010. Currently, deep learning is becoming the third artificial intelligence boom.

Deep learning is a superposition of neural networks and has deep structure. Deep learning has more four layers include input layer and output layer. It is repeated learning by the time and the input data are gradually being transferred to the deeper and deeper from the first layer. Previously, researchers and engineers had set the parameters manually. Now, that deep learning automatically learns valid features of discrimination. It has been improved accuracy of pattern recognition. It is attracting attention.

In June 2012, it was able to recognize the cat on computer by research on Google. After that, it comes to be used in various fields. For example, image field, emotional awareness, module of self-driving, medical field, financial field and so on [1]. Google released a software library of machine learning techniques including deep learning in 2015. We raised interest in deep learning. Figure 1 shows recognition of handwritten numbers by deep learning with Google's software library [2]. We can see that number are learned from the left in order. It takes the character of the number and judges which number it is.

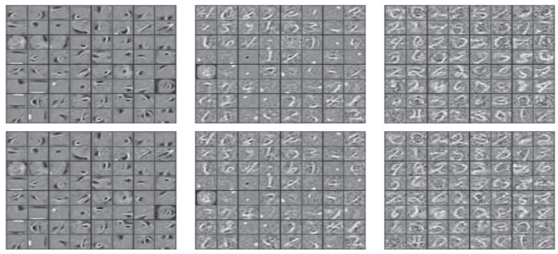

Figure 1: Recognition of handwritten numbers by deep learning.

In deep learning, there are a variety of techniques in order to improve the learning accuracy. Technique of back propagation compares input data and output data to find deviations of feature values and optimize weights [3]. This technique can reduce learning loops. If the learning period is too long or the training data is not typical, it adapts to features unrelated to the features that should be originally learned. This phenomenon is called over learning. In such a case, it uses the method of dropout to correspond to input data. Deep learning requires a large amount of learning data. When there are not enough input data, input data are shifted slightly due to use noise to increase features. There are various techniques like these.

In this study, we propose a system of deep learning. This system is noisy back propagation. Weights of neurons are optimized using a noise in the back propagation. The purposes of this study are to improve learning accuracy and to reduce learning loops. We compare proposed system with the conventional system using two data sets.

2. Proposed method

Figures 2 and 4 show the conventional systems. Figures 3 and 5 show the proposed systems. Arrows on left to right (h_1, h_2) indicate propagation of input signal and right to left (h_{1back}, h_{2back}, $\mathrm{y_{back}}$) indicate learning of back propagation. x, y and w show input data, output data and weight between neurons.

(A) Iris data set [4]

The iris data set is data to classify into 3 types of iris with

4 features. There is a total of 150 data set. We use a model consisting of two layers of intermediate layers and Gaussian distribution (0, 0.01) for noise and add it to equations (2). In addition, the putting place of noise is changed. We experiment 8 patterns for the conventional system without noise, (h_{1back}), (h_{2back}), $(\mathrm{y_{back}})$, $(h_{1back}\ h_{2back})$, $(h_{1back}\ \mathrm{y_{back}})$, $(h_{2back}\ \mathrm{y_{back}})$ and $(h_{1back}\ h_{2back}\ \mathrm{y_{back}})$. These combinations indicate where noise is added. We define the learning loops = 25000 and the number of learning data sets = 150.

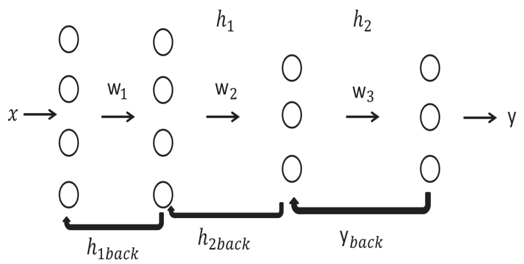

Figure 2: A schematic drawing of deep learning consisting of two intermediate layers.

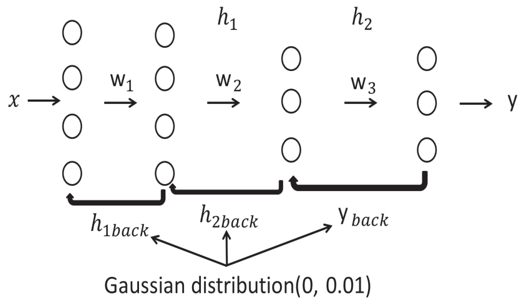

Figure 3: A schematic drawing of proposed deep learning consisting of two intermediate layers.

$$\begin{cases} h_1 = \dfrac{1}{1+e^{(-\sum(\mathrm{w}_1 x))}} \\ h_2 = \dfrac{1}{1+e^{(-\sum(\mathrm{w}_2 h_1))}} \\ x = \dfrac{1}{1+e^{(-\sum(\mathrm{w}_3 h_2))}} \end{cases} \tag{1}$$

$$\begin{cases} \mathrm{y_{back}} = (\mathrm{y} - t\mathrm{y})(1-\mathrm{y})\mathrm{y} \\ h_{2back} = \sum \mathrm{w}_3 \mathrm{y_{back}}(1-h_2)h_2 \\ h_{1back} = \sum \mathrm{w}_2 h_{2back}(1-h_1)h_1 \end{cases} \tag{2}$$

$$\begin{cases} \mathrm{w}_1^{(\mathrm{l}+1)} = \mathrm{w}_1^{\mathrm{l}} - \varepsilon x h_{1back} \\ \mathrm{w}_2^{(\mathrm{l}+1)} = \mathrm{w}_2^{\mathrm{l}} - \varepsilon h_1 h_{2back} \\ \mathrm{w}_3^{(\mathrm{l}+1)} = \mathrm{w}_3^{\mathrm{l}} - \varepsilon h_2 \mathrm{y_{back}} \end{cases} \tag{3}$$

ε is the rate of decay, which is 0.01. h means activation function. w means weights between neurons. The initial value of the weights is chosen by random from 0.04 or less.

(B) Cars data set [4]

The cars data set is five items of car price and safety as input, and outputs as four evaluations. There is a total of 1728 data set. We use a model consisting of three layers of intermediate layers and Gaussian distribution (0, 0.01) for noise and add it to equations (5). In addition, the putting place of noise is changed. We experiment 15 patterns for the conventional system without noise, (h_{1back}), (h_{2back}), (h_{3back}), $(\mathrm{y_{back}})$, $(h_{1back}\ h_{2back})$, $(h_{1back}\ h_{3back})$, $(h_{1back}\ \mathrm{y_{back}})$, $(h_{1back}\ h_{2back}\ h_{3back})$, $(h_{1back}\ h_{2back}\ \mathrm{y_{back}})$, $(h_{1back}\ h_{3back}\ \mathrm{y_{back}})$, $(h_{1back}\ h_{2back}\ h_{3back}\ \mathrm{y_{back}})$, $(h_{2back}\ h_{3back})$, $(h_{2back}\ \mathrm{y_{back}})$, $(h_{2back}\ h_{3back}\ \mathrm{y_{back}})$ and $(h_{3back}\ \mathrm{y_{back}})$. These combinations indicate where noise is added. We define as the learning loops = 1000 and the number of learning data sets = 1728.

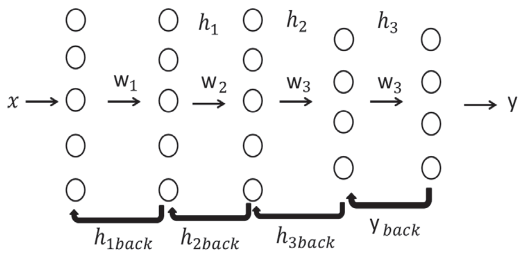

Figure 4: A schematic drawing of deep learning consisting of three intermediate layers.

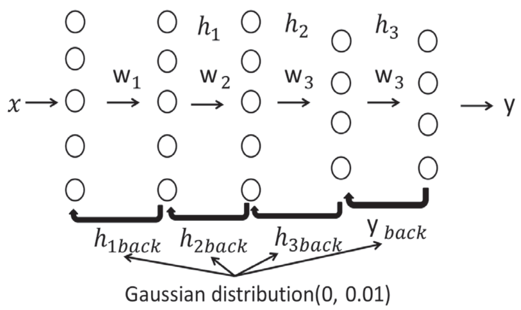

Figure 5: A schematic drawing of proposed deep learning consisting of three intermediate layers.

$$\begin{cases} h_1 = \dfrac{1}{1+e^{(-\sum(\mathrm{w}_1 x))}} \\ h_2 = \dfrac{1}{1+e^{(-\sum(\mathrm{w}_2 h_1))}} \\ h_3 = \dfrac{1}{1+e^{(-\sum(\mathrm{w}_3 h_2))}} \\ \mathrm{y} = \dfrac{1}{1+e^{(-\sum(\mathrm{w}_4 h_3))}} \end{cases} \tag{4}$$

$$\begin{cases} \mathrm{y}_{\mathrm{back}} = (\mathrm{y} - ty)(1 - \mathrm{y})\mathrm{y} \\ h_{3back} = \sum \mathrm{w}_4 \mathrm{y}_{\mathrm{back}} (1 - h_3) h_3 \\ h_{2back} = \sum \mathrm{w}_3 h_{3back} (1 - h_2) h_2 \\ h_{1back} = \sum \mathrm{w}_2 h_{2back} (1 - h_1) h_1 \end{cases} \tag{5}$$

$$\begin{cases} \mathrm{w}_1{}^{(l+1)} = \mathrm{w}_1{}^{l} - \varepsilon x h_{1back} \\ \mathrm{w}_2{}^{(l+1)} = \mathrm{w}_2{}^{l} - \varepsilon h_1 h_{2back} \\ \mathrm{w}_3{}^{(l+1)} = \mathrm{w}_3{}^{l} - \varepsilon h_2 h_{3back} \\ \mathrm{w}_4{}^{(l+1)} = \mathrm{w}_4{}^{l} - \varepsilon h_3 \mathrm{y}_{\mathrm{back}} \end{cases} \tag{6}$$

ε is the rate of decay, which is 0.4. h means activation function. w means weights between neurons. The initial value of the weights is chosen by random from 0.3 or less.

3. Simulation result

(A) Iris data set

Figure 6 shows the change of the learning accuracy and learning loops of a conventional system and proposed system. Table 1 shows minimum learning accuracy for all kinds of patterns. Table 2 shows learning loops when learning accuracy is 0.1 or less.

From Figure 6, the learning accuracy and learning loops by using Gaussian distribution (0, 0.01) are better than not using it. From Table 1, most results by using the noise are found to be better.

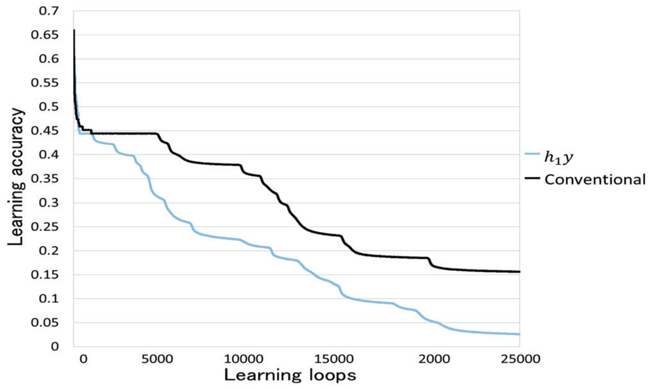

Figure 6: The simulation results of a schematic drawing of deep learning consisting of two intermediate layers.

Table 1: Learning accuracy of model (A).

	accuracy
Conventional	0.155788
h_{1back}	0.074552
h_{2back}	0.094836
$\mathrm{y}_{\mathrm{back}}$	0.099584
$h_{1back}h_{2back}$	0.076192
$h_{1back}\mathrm{y}_{\mathrm{back}}$	0.025454
$h_{2back}\mathrm{y}_{\mathrm{back}}$	0.119983
$h_{1back}h_{2back}\mathrm{y}_{\mathrm{back}}$	0.136539

Table 2: Learning loops of model (A).

	loops
h_{1back}	16878
h_{2back}	24668
$\mathrm{y}_{\mathrm{back}}$	24984
$h_{1back}h_{2back}$	22974
$h_{1back}\mathrm{y}_{\mathrm{back}}$	15599

(B) Cars data set

Figure 7 shows the learning accuracy and learning loops of a conventional system and proposed systems. Table 3 shows learning accuracy of minimum value of all kinds of patterns. Table 4 shows learning loops for all kinds of patterns when learning accuracy is 0.064 or less.

From Figure 7, the learning accuracy and learning loops by using Gaussian distribution (0, 0.01) are better than not using it.

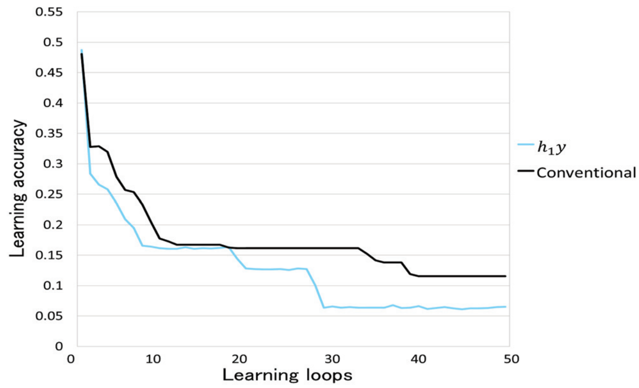

Figure 7: The simulation results of a schematic drawing of deep learning consisting of three intermediate layers.

Table 3: Learning accuracy of model (B).

	minimum
Conventional	0.063878
h_{1back}	0.064484
h_{2back}	0.06359
h_{3back}	0.067046
$\mathrm{y}_{\mathrm{back}}$	0.059693
$h_{1back}h_{2back}$	0.063429
$h_{1back}h_{3back}$	0.066899
$h_{2back}h_{3back}$	0.06809
$h_{1back}h_{2back}h_{3back}$	0.067258
$h_{1back}\mathrm{y}_{\mathrm{back}}$	0.060497
$h_{2back}\mathrm{y}_{\mathrm{back}}$	0.06114
$h_{3back}\mathrm{y}_{\mathrm{back}}$	0.066832
$h_{1back}h_{2back}\mathrm{y}_{\mathrm{back}}$	0.080702
$h_{1back}h_{3back}\mathrm{y}_{\mathrm{back}}$	0.064958
$h_{2back}h_{3back}\mathrm{y}_{\mathrm{back}}$	0.072979
$h_{1back}h_{2back}h_{3back}\mathrm{y}_{\mathrm{back}}$	0.082566

Table 4: Learning loops of model (B).

	loops
$y_{\rm back}$	427
$h_{1back}y_{\rm back}$	338
$h_{2back}y_{\rm back}$	410

4. Conclusion

In this study, we proposed technique of deep learning. It is using noise at back propagation. In the proposed technique, we choose Gaussian distribution (0, 0.01) for noise. Then, we examined whether this technique is effective in simulation that is using two data sets.

In simulations, we were able to improve learning accuracy and reduce learning loops to use noise at back propagation. From the tendency of research results of two data sets, we found that results are improved by using noise at back propagation from the output layer to the final intermediate layer. Because back propagation of the output layer is compared with input data and output data, so it is properly carried out weight adjustment.

In the future work, we will use other data sets to demonstrate usability and a variety of noise. Then, we obtain the best performance of the proposed technique.

Acknowledgment

This work was partly supported by JSPS Grant-in-Aid for Scientific Research 16K06357.

References

[1] Y. Iwatani and H. Furusho, "Lane Recognition by Deep Learning for Autonomous Driving on City Roads," Proceedings of Nissan technical review, pp. 40 - 46, 2016

[2] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network," Proceedings of Advances in Neural Information Processing Systems, pp. 396 - 404, 1990

[3] Rumelhart, E. David, E. G. Hinton and J. R. Williams, "Learning representations by back-propagating errors," Cognitive modeling 5.3, 1988

[4] A. Asuncion and D. Newman (2007), "Machine Learning Repository," https://archive.ics.uci.edu/ml/about.html