15-5

Feedforward Neural Networks with Threshold for Avoiding Over-Training

Akihiro ITAGAKI[1], Ryuta YOSHIMURA[1], Yoko UWATE[1] and Yoshifumi NISHIO[1]
([1] Tokushima University)

1. Introduction

Neural Network is a model imitating the human brain. It is widely used in various areas. Feedforward back propagation which is well known to solve social problem is one of the Neural Networks. When learning is repeated for all training data, over learning occurs. We keep network from over training by setting the threshold. If the error value falls below the threshold, training data would not be used for learning.

2. Proposed method

Proposed system that is used in this study is shown in Fig.1. The right arrow indicates the propagation of the input signal. The reverse arrow indicates the back propagation in Fig.1. In this study, the error value in the training data is not used for learning below the set value. Thereby it can prevent the problem of over training. This proposed system can reduce error value.

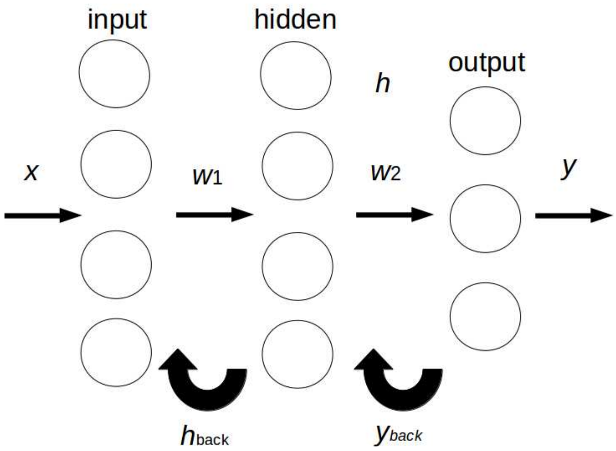

Figure 1: Feedforward Newral Network.

$$\begin{cases} h = \dfrac{1}{1+e^{(-\sum(w_1 x))}} \\ y = \dfrac{1}{1+e^{(-\sum(w_2 h))}} \end{cases} \quad (1)$$

Equation (1) is the sigmoid functions. It is used in the output of each neuron and calculation in back propagation. h means propagation. x means input. y means output. w means weights between neurons.

$$\begin{cases} y_{back} = (y - ty)(1-y)y \\ h_{back} = \sum w y_{back}(1-h)h \end{cases} \quad (2)$$

$$\begin{cases} {w_1}^{(l+1)} = {w_1}^{l} - \varepsilon\ x h_{back} \\ {w_2}^{(l+1)} = {w_2}^{l} - \varepsilon\ h y_{back} \end{cases} \quad (3)$$

h_{back} and y_{back} are back probagation. ε is the rate of decay which is 0.01. The initial value of the weights is chosen by random from 0.01 to 0.04.

3. Simulation result

We define as the number of input layer = 4, the number of hidden layer = 4, the number of layer = 3, the learning loops = 100000, training data sets = 5 and test data sets =48. Iris data set is used for training data. The test data is larger than the training data because it causes over training. The threshold is set to 7 patterns and error value is shown in Table 1. The error in this is an error in the test data.

The error value is the average of 20 simulations. In case of $error < 0.3$, the result is the smallest. However, in case of $error < 1.0$, error value becomes larger than when the threshold value is not set. If you set the threshold too high, the learning loops will decrease significantly. As a result, when we set the threshold $error < 0.3$, the best result is obtained.

Table 1: Error value

threshold	error value
no threshold	0.445
$error < 0.01$	0.439
$error < 0.05$	0.409
$error < 0.1$	0.393
$error < 0.3$	0.391
$error < 0.5$	0.428
$error < 1.0$	0.466

4. Conclusion

As the simulation results, we found that setting the threshold reduces over training. Therefore the threshold reduces the error value. Especially in case of $error < 0.1$ and $error < 0.3$, the proposed method obtains good results.

In the future work, we would like to obtain better results. Therefore, we would like to consider a new proposed method such as adding noise to data. Furthermore, We think that we will try to reduce error value by deep learning which increased the hidden layer.

Reference

[1]Isao Taguchi and Yasuo sugai, "The Error Territory in the Functional Learning of the Three-Layer Networks", Keiai University International research. vol. 10, pp.63-67, 2002.

[2]Jacques de Villiersand Etienne Barnard, "Backpropagation Neural Nets with One and Two Hidden Layers", IEEE Transactions on Neural Networks. vol. 4. no.1 January 1992, pp. 136-141, 1992.