Error Back-Propagation Algorithm with Independent Component Analysis

Sang-Hoon Oh*
*Mokwon University, Korea
E-mail : shoh@mokwon.ac.kr

1. Introduction

Feed-forward neural networks(FNNs) are universal approximators, which can approximate any function with enough number of hidden nodes[1]. Usually the error back-propagation(EBP) algorithm using mean-squared error is used to train FNNs[2]. However, the EBP algorithm suffers from slow convergence of learning. In order to improve the performance of EBP algorithm, various cost functions such as CE(cross-entropy)[3], nCE(n-th order extension of CE)[4], and CFM(classification figure of merit)[5] have been proposed.

There have been many attempts to construct the structure of FNNs for performance improvement. Usually, in pattern classification applications, we allocate one output node per class and the index of output node with maximum value represents the classified class. Increasing the number of output nodes per class attained better classification performance than one output node per class[6]. However, this strategy could not guarantee the performance improvement. Increasing the number of output nodes per class results in intensifying the dependency among output nodes and this will degrade the classification performance of FNNs.

In this paper, we propose a new algorithm to make the output nodes as independent as possible. This is the merge of EBP[2] and ICA(independent component analysis)[7] algorithms.

2. Feed-Forward Neural Networks and Additional Output Nodes for Performance Improvement

Consider a FNN consisting of N inputs, H hidden, and M output nodes. When a training pattern $\mathbf{x}=[x_1,x_2,\cdots,x_N]$ is presented to the FNN, the j-th hidden node is given by

$$h_j \equiv h_j(\mathbf{x}) = \tanh(\hat{h}_j/2) \text{ where } \hat{h}_j = \sum_{i=0}^{N} w_{ji}x_i,\ j=1,2,\cdots,H. \tag{1}$$

Here, $x_0=1$ and w_{ji} denotes the weight connecting the i-th input x_i to h_j. The k-th output node is

$$y_k \equiv y_k(\mathbf{x}) = \tanh(\hat{y}_k/2) \text{ where } \hat{y}_k = \sum_{j=0}^{H} v_{kj}h_j,\ k=1,2,\cdots,M. \tag{2}$$

Also, $h_0=1$ and v_{kj} denotes the weight connecting h_j to y_k.

Let desired output vector corresponding to the training pattern $\mathbf{x}$ be $t=[t_1,t_2,\cdots,t_M]$, where the class from which $\mathbf{x}$ originates is coded as follows:

$$t_k = \begin{cases} +1, & \text{if } \mathbf{x} \text{ originates from class } k, \\ -1, & \text{otherwise.} \end{cases} \tag{3}$$

We call y_k the target node of class k. The conventional error function for P training patterns is $E=\sum_{p=1}^{P}\sum_{k=1}^{M}\left(t_k^{(p)}-y_k^{(p)}\right)^2$ EBP algorithm minimizes E through iterative update of weights to the negative direction of the error function.

In order to prove the effectiveness of increasing the number of output nodes, we model a problem with two classes, c_1 and c_2, with their prior probabilities $P(c_1)=P(c_2)$. Also, we assume that the outputs of FNNs trained to resolve this problem have uniform distributions in each class. The targets for outputs are given by -1 for class 1 and +1 for class 2. Here, c_1 and c_2 denote the class 1 and 2, respectively. Let us consider the one-output node case for the two-class problem. The uniform distribution of output z in each class is assumed as

$$p(z \mid c_1) = \begin{cases} \dfrac{1}{-4z^{(c_1)}}, & 3z^{(c_1)} < z < -z^{(c_1)} \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

and

$$p(z \mid c_2) = \begin{cases} \dfrac{1}{4z^{(c_2)}}, & -z^{(c_2)} < z < 3z^{(c_2)} \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where $p(.)$ is a pdf (probability density function). Also, $z^{(c_1)}$ is negative and $z^{(c_2)}$ is positive. For simplicity, we use $z^{(c_2)} = -z^{(c_1)} = z^{(c)} > 0$. Using Bayes' decision rule for many classes, we decide the classification result as

$$j = \max_j \, p(z \mid c_j)P(c_j). \tag{6}$$

Here, we consider a two-class problem whose targets are coded as -1 for class 1 and +1 for class 2 with the same prior probabilities. Then we can decide that an input pattern belongs to class 1 if $z < 0$ and it belongs to class 2 otherwise. We can derive the misclassification probability as

$$P_E = \int_{-z^{(c)}}^{0} p(z \mid c_2)P(c_2)dz + \int_{0}^{z^{(c)}} p(z \mid c_1)P(c_1)dz = \frac{1}{4}. \tag{7}$$

3. Proposed Algorithm

In order to make the output nodes as independent as possible, we adopt the ICA algorithm[7], which can work in the environment that the number of output nodes is different with the number of inputs in the ICA network. Since the proposed algorithm is only for the last output layer, the weights in hidden layers are the same with the EBP algorithm. In the last output layer, the weights are updated as follows:

$$\Delta \mathbf{V} = \eta_1 [\delta_k][h_j]^T \tag{8}$$

$$\Delta \mathbf{V}^c = \eta_2 [\mathbf{I}^c + f(\hat{\mathbf{y}}_c)\hat{\mathbf{y}}_c^T]\mathbf{V}^c \tag{9}$$

Eq. (8) is from the EBP algorithm[2] to reduce the distance between the output and desired values. Eq. (9) is from the ICA algorithm[7] to make output nodes independent.

4. Conclusion

In order to improve the classification performance of FNNs, we proposed a new algorithm which can make output nodes as independent as possible.

5. References

[1] K. Hornik, M. Stincombe, and H. White, "Multilayer feed-forward networks are universal approximators," Neural Networks, vol. 2, 1989, pp. 359-366.
[2] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*, MIT Press, Cambridge, MA(1986).
[3] A. van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," Neural Networks, vol. 4, 1992, pp. 465-471.
[4] S.-H. Oh, "Improving the error back-propagation algorithm with a modified error function, " IEEE Trans. Neural Networks, vol.8, 1997, pp. 799-803.
[5] J. B. Hampshire II and A. H. Waibel, "A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Networks. IEEE Trans. Neural Networks, vol. 1, 1990, pp. 216—218.
[6] S.-H. Oh, "Performance Improvement of Multilayer Perceptrons with Increased Output Nodes," J. of the Korea Contents Association, vol.9, no.1, pp. 123-130, 2009.
[7] M. Girolami, A. Cichocki, and S.-I. Amari, "A Common Neural-Network Model for Unsupervised Exploratory Data Analysis and Independent Component Analysis," IEEE Trans. Neural Networks, vol.9, no.6, pp. 1495-1501, 1998.