Hidden Node Pruning of Multilayer Perceptrons Based on Redundancy Reduction

Sang-Hoon Oh

Mokwon University, Doan-dong, Seo-gu, Daejon, Korea
shoh@mokwon.ac.kr

Abstract. Among many approaches to choosing the proper size of neural networks, one popular approach is to start with an oversized network and then prune it to a smaller size so as to attain better performance with less computational complexity. In this paper, a new hidden node pruning method is proposed based on the redundancy reduction among hidden nodes. The redundancy information is given by correlation coefficients among hidden nodes and this can save computational complexity. Experimental results demonstrate the effectiveness of the proposed method.

Keywords: Multilayer perceptron, hidden node pruning, redundancy reduction.

1 Introduction

When applying MLPs (multilayer perceptrons) to real problems, it is very important to find a reasonable number of hidden nodes for good generalization performance. An MLP with too many hidden nodes may accurately fit the training data, but may have bad generalization performance due to over-fitting of the training data. On the other hand, an MLP with too small size may save on computational costs, but may have insufficient processing elements to approximate the true function.

Among many approaches to find the adequate size of MLP to solve a real problem, the pruning approach [1][2][3] appears reasonable since it can more easily avoid local minima during training and have a good generalization performance after completion of pruning [4]. Although sensitivity analyses are good criterion for pruning methods [3][4][5], these are with high computational complexity. On the contrary, pruning methods based on the role interpretation of hidden nodes are efficient with less computational complexity [6][7]. In this sense, this paper adopts the second strategy and proposes a new pruning method based on the redundancy among hidden nodes.

2 Proposed Method

For pattern recognition applications of MLPs, the hidden nodes are trained to do a role of feature extraction from input patterns. This is done by linear projections onto hidden weight vectors followed by element-wise sigmoidal transformations. After successful training of MLPs, hidden weight vectors tend to be near orthogonal and hidden nodes are less correlated. If hidden nodes are highly correlated, there must be

G. Lee, D. Howard, and D. Ślęzak (Eds.): ICHIT 2011, LNCS 6935, pp. 245–249, 2011.

a redundancy among the hidden nodes and we can remove one hidden node without a serious impact on performance degradation.

Let's derive the correlation coefficient among hidden nodes. Consider an MLP with *N* inputs, *H* hidden nodes and *M* output nodes [8]. The weighted sum to the *i*th hidden node is given by

$$\hat{h}_i = w_{i0} + \sum_{k=1}^{N} w_{ik} x_k, \; i = 1,2,...,H, \tag{1}$$

where w_{ik} is the weight connecting the *k*th input to the *i*th hidden node and w_{i0} is the bias. Under the assumption that x_k's $(k = 1,2,\ldots,N)$ are zero-mean i.i.d. (independent, identically distributed) random variables with the standard deviation σ, the expectations are

$$E[\hat{h}_i] = w_{i0} \text{ and } E[\hat{h}_i \hat{h}_j] = w_{i0} w_{j0} + \sigma^2 \sum_{k=1}^{N} w_{ik} w_{jk} \,. \tag{2}$$

Here, $E[\cdot]$ is the expectation operator. Also, the variance is

$$\sigma_{\hat{h}_i}^2 \equiv E[\hat{h}_i^2] - E^2[\hat{h}_i] = \sigma^2 \sum_{k=1}^{N} w_{ik}^2 . \tag{3}$$

The correlation coefficient between $\hat{h}_i$ and $\hat{h}_j$ is given by

$$Cor[\hat{h}_i, \hat{h}_j] \equiv \frac{E[\hat{h}_i \hat{h}_j] - E[\hat{h}_i] E[\hat{h}_j]}{\sigma_{\hat{h}_i} \sigma_{\hat{h}_j}} = \frac{\sum_{k=1}^{N} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{N} w_{ik}^2} \sqrt{\sum_{k=1}^{N} w_{jk}^2}} \,. \tag{4}$$

Because of the zero-mean i.i.d. assumption of inputs, $Cor[\hat{h}_i, \hat{h}_j]$ can be easily calculated using hidden weight vectors. If the hidden weight vectors are orthogonal, $Cor[\hat{h}_i, \hat{h}_j] = 0$.

So, the pruning algorithm for reducing the redundancy among hidden nodes is proposed as follows:

① Initialize an MLP with enough hidden nodes.
② Train the MLP and check the performance for test patterns.
③ If there is no performance improvement for test patterns after additional 50 epochs, remove one hidden node which has the largest $\left| Cor[\hat{h}_i, \hat{h}_j] \right|$.
④ Go to step 2.

3 Simulations

To verify the effectiveness of the proposed method, a hand-written digit recognition (HDR) task was simulated. A total of 18,468 handwritten digitized images from the

CEDAR database [9] are used for training after size normalization. A digit image consists of 12 x 12 pixels and each pixel takes on integer values from 0 to 15. The MLP with 144 inputs, 70 hidden nodes and 10 output nodes was initialized with weights randomly selected from the uniform distribution on $[-1\times10^{-4}, 1\times10^{-4}]$. The MLP was trained using the error back-propagation algorithm with the nth order error function (n=4) [10] and the hidden nodes were pruned using the proposed method. The value of learning rate was 0.005.

Fig. 1(a) shows the misclassification ratios of training and test patterns, which were evaluated according to the Max. rule. Fig. 1(b) shows the number of hidden nodes during training. At the 2140 epoch, the MLP with 48 hidden nodes shows the best performance for test patterns. From Figs. 1(a) and 1(b), we can say that the proposed method successfully reduced the number of hidden nodes.

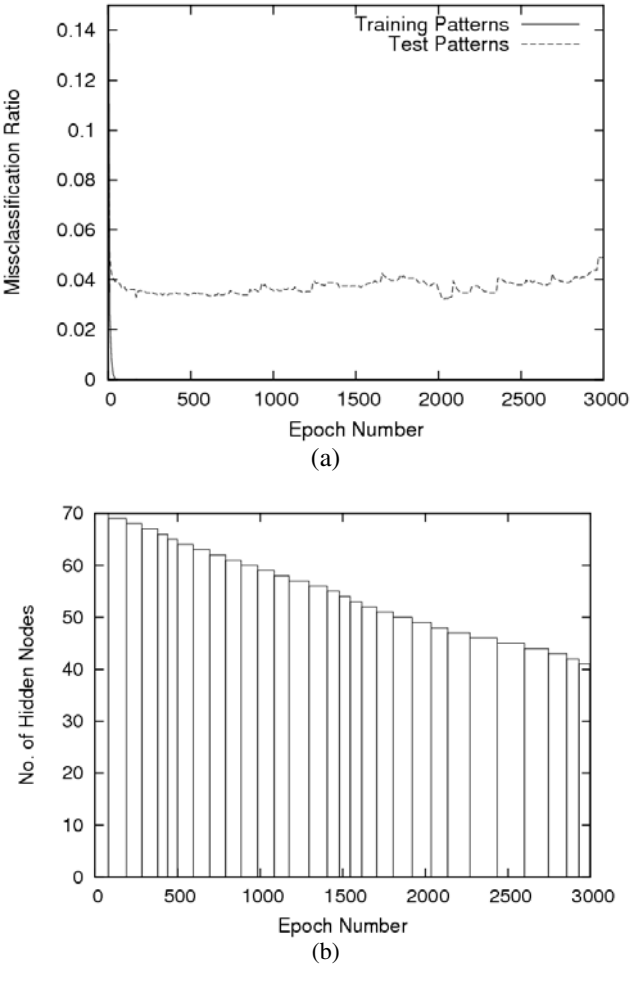

Fig. 1. Simulation results. (a) Misclassification ratios vs. training epochs. (b) The number of hidden nodes vs. training epochs.

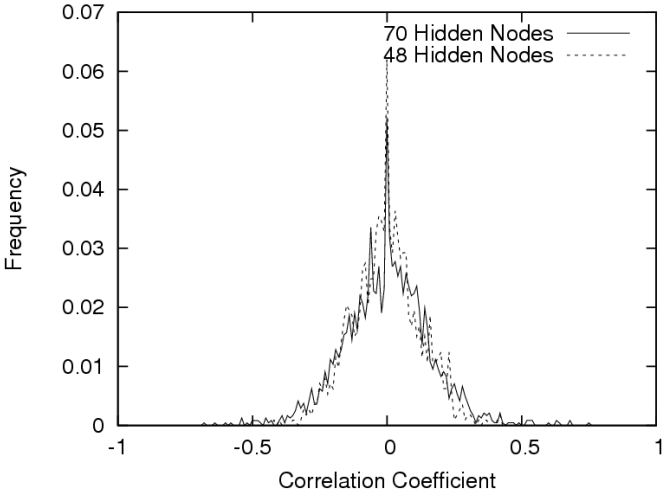

Fig. 2. Histogram of correlation coefficients among hidden nodes

Secondly, we investigated the redundancy among hidden nodes. During training with the proposed pruning algorithm, $Cor[\hat{h}_i, \hat{h}_j]$ was evaluated at the cases of 70 and 48 hidden nodes. Their histograms were drawn as shown in Fig. 2. Compared with the case of 70 hidden nodes, $\left|Cor[\hat{h}_i, \hat{h}_j]\right|$ with the case of 48 hidden nodes decreased and concentrated on zero value. When the training epochs increased and the number of hidden nodes decreased, $Cor[\hat{h}_i, \hat{h}_j]$ approached to a small value. Thus, hidden nodes tend to be less correlated after successful training. Also, by the central limit theorem, $[\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_H]^T$ can be approximated to a jointly Gaussian vector if x_k's $(k = 1,2,\ldots,N)$ are i.i.d. random variables [11]. The independence among $\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_H$ can be implied by their uncorrelatedness. So, we can argue that hidden nodes extract independent features as far as possible.

Since the pruning is based on the hidden weight vectors given by Eq. (4), it consumes approximately NH^2 multiplications to find a hidden node to be pruned. On the contrary, about $(3M+2) \times P$ multiplications are necessary in the pruning method based on the sensitivity information [3]. Here, P is the number of training patterns. In real problems, P should be very large and the discrepancy will increase. In the HDR simulation, the proposed method can save 39% of multiplications compared to the pruning method based on the sensitivity information [3].

4 Conclusions

This paper proposed a new pruning method based on the redundancy among hidden nodes. The redundancy information was given by the correlation coefficients among hidden nodes. Through simulations of HDR problem, we verified that the proposed

method successfully reduced the number of hidden nodes and attained a good performance for test patterns. Also, the proposed method could save computational complexity compared with the pruning method based on the sensitivity information.

References

1. Xu, J., Ho, D.W.C.: A new training and pruning algorithm based on node dependence and Jacobian rank deficiency. Neurocomputing 70, 544–558 (2006)
2. Zhang, L., Jiang, J.-H., Liu, P., Liang, Y.-Z., Yu, R.-Q.: Multivariate nonlinear modeling of fluorescence data by neural network with hidden node pruning algorithm. Analytica Chimica Acta 344, 29–39 (1997)
3. Engelbrecht, A.P.: A new pruning heuristic based on variance analysis of sensitivity information. IEEE Trans. Neural Networks 12, 1386–1399 (2001)
4. Zeng, X., Yeung, D.S.: Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity measure. Neurocomputing 69, 825–837 (2006)
5. Lauret, P., Fock, E., Mara, T.A.: A node pruning algorithm based on a Fourier amplitude sensitivity test method. IEEE Trans. Neural Networks 17, 273–293 (2006)
6. Sietsma, J., Dow, R.J.F.: Creating artificial neural networks that generalize. Neural Networks 4, 67–79 (1991)
7. Hagiwara, M.: Removal of hidden units and weights for back propagation networks. In: Int. Joint Conf. Neural Networks, pp. 351–354 (1993)
8. Rumelhart, D.E., McClelland, J.L.: Parallel Distributed Processing. MIT Press, Cambridge (1986)
9. Hull, J.J.: A database for handwritten text recognition research. IEEE Trans. Pat. Ana. Mach. Int. 16, 550–554 (1994)
10. Oh, S.-H.: Improving the error back-propagation algorithm with a modified error function. IEEE Trans. Neural Networks 8, 799–803 (1997)
11. Lee, Y., Oh, S.-H., Kim, M.W.: An analysis of premature saturation in back propagation learning. Neural Networks 6, 719–728 (1993)