Merging of Two Artificial Neural Networks[1]

Mun Hyuk Kim and Jin Young Choi
School of Electrical Engineering and Computer Science, Seoul National Univ.,
ENG420-048. San 56-1, Shilim-dong, Kwanak-ku, Seoul, Korea. 151-744
Tel: +82-2-872-7283, Fax: +82-2-885-4459
e-mail : mhkim@neuro.snu.ac.kr
School of Electrical Engineering and Computer Science, Seoul National Univ.,
ENG420-048. San 56-1, Shilim-dong, Kwanak-ku, Seoul, Korea. 151-744
Tel: +82-2-872-7283, Fax: +82-2-885-4459
e-mail : jychoi@ee.snu.ac.kr

Abstract: This paper addresses the problem of merging two feedforward neural networks into one network. Merging is accomplished at the level of hidden layer. A new network selects its hidden layer' s units from the two networks to be merged. We uses information theoretic criterion (quadratic mutual information) in the selection process. The hidden unit' s output and the target patterns are considers as random variables and the mutual information between them is calculated. The mutual information between hidden units are also considered to prevent the statistically dependent units from being selected. Because mutual information is invariant under linear transformation of the variables, it shows the property of the robust estimation.

1. Introduction

According to the principle of divide and conquer, there have been many tries to solve a complex problem by dividing it into a number of simple tasks. The solutions to those tasks are combined to make the solution of the complex problem. In the field of artificial neural networks, committee machines and data fusions can be considered as those methods. These methods combine each expert's output to produce a better result.

Committee machines [1] are comprised of experts whose functions are the same. Only the method of distributing the data set to each expert can be various. In the committee machine based on ensemble average, all experts are trained with the same data set. On the other hand, in the method based on boosting, an expert can have its own distribution of training data. In data fusion [2], the system can combines experts with different sources. They do not share the common inputs and can have different kind of input patterns.

In the process of knowledge integration, these methods just combine the outputs of the experts. So the size of resulting overall network is approximately proportional to the number of the experts. Because the information merging is executed at the level of the outputs, it is not easy to utilize the information acquired by each expert for another but similar task.

In this paper, we propose a method to merge two feedforward neural networks into one. Two networks should share the common inputs and can have different kind of outputs. The merging process is picking up useful hidden units from two networks to be merged. The usefulness is measured by information theoretic measure – mutual information (entropy).

Mutual information is invariant under linear transformations of the variables. On the contrary, linear scaling can modify the result of the linear transformation based methods such as PCA. In addition, methods based on linear dependence, like correlation, cannot take care of the arbitrary relations between the pattern coordinates and the different classes. On the other hand, mutual information can measure arbitrary relations between variables [3].

With our method, the level of information merging is not confined to the output level. If two networks do the same function as in the case of committee machine, the integrated network can outperform its parent without being double-sized. If the networks do different functions from each other, we can make a new network that efficiently learns some similar - not the same - tasks by utilizing all the information its parents have.

This paper is organized as follows: In the second section, previous approaches to merging feedforward neural networks are introduced. In section 3, we introduce a method of estimating the quadratic mutual information with discrete data set. Upon this estimation, network merging method using quadratic mutual information is proposed. Finally, some simulation results are given in section 4.

2. Previous Approaches

Bahrami [4] integrated two multilayer perceptrons (Net1 & Net2) for classification of two sets of data. Each network has 2-4-4-2 structure. Net1 can discriminate the letter 'A' from 'B', and Net2 can tell the letter 'a' from 'b'. The object of the experiment is making a new network that classifies ('A','a') into one set and ('B','b') into the other. According to his experiments, the network which simply contains all the weights of the two networks (2-8-8-2 structure), shows the best result. In this case, only the weights of the output layer were re-calculated by linear method. This experiment and Lo & Bassu's result [5] implies that it is very important to properly locate the hidden units' decision boundaries.

Burrascano and Pirollo [6] made use of the probabilistic interpretation of neural modeling. They estimated the weight' s probability distribution of the new merged network from those of the two networks to be merged.

[1] This research work was supported by the Braintech 21 project, the Brain Korea 21 project, and the Super intelligence project, in Korea.

Their experimental result shows that the merged network' s performance is better than those of its parent networks. But this result relies on the consideration that merged weight falls in a neighborhood of the segment linking the weights of its parents.

3. Merging of Two Networks

In the merging problem, it is supposed that we already have plenty of properly located hidden units in the two networks to be merged. All we have to do is combining them without magnifying the size of the integrated network.

3.1 Quadratic Mutual Information

Mutual information is known as a good measure that indicates the arbitrary dependencies between random variables. But, it is not easy to estimate Shannon' s definition of mutual information from discrete set of data.

Principe etc. [7] introduced a non-parametric approach to estimate the information from a discrete set of data. They proposed the integration of the Cauchy-Schwartz distance and an Euclidean difference with the Parzen window method [8] with Gaussian kernel. Parzen window method estimates PDF of a random variable. With Gaussian kernel, PDF of a variable and joint PDF can be estimated as:

$$\hat{f}_Y(z)=\frac{1}{N}\sum_{i=1}^{N}G(z-y_i,\sigma^2 I)$$

$$\hat{f}_{Y_1Y_2}(z_1,z_2)=\frac{1}{N}\sum_{i=1}^{N}G(z_1-y_{i1},\sigma^2 I)G(z_2-y_{i2},\sigma^2 I)\cdot$$

The resulting Cauchy-Schwartz quadratic mutual information is

$$I_{CS}(Y_1,Y_2)=\log\frac{\int\int f_{Y_1Y_2}(z_1,z_2)^2\,dz_1dz_2\int\int f_{Y_1}(z_1)^2 f_{Y_2}(z_2)^2\,dz_1dz_2}{\left(\int\int f_{Y_1Y_2}(z_1,z_2)f_{Y_1}(z_1)f_{Y_2}(z_2)dz_1dz_2\right)^2},$$

and Euclidean difference quadratic mutual information is

$$I_{ED}(Y_1,Y_2)=\int\int f_{Y_1Y_2}(z_1,z_2)^2dz_1dz_2+\int\int f_{Y_1}(z_1)^2 f_{Y_2}(z_2)^2dz_1dz_2 -2\int\int f_{Y_1Y_2}(z_1,z_2)f_{Y_1}(z_1)f_{Y_2}(z_2)dz_1dz_2$$

Each element of the mutual information can be calculated as following:

$$\int\int f_{Y_1Y_2}(z_1,z_2)^2dz_1dz_2\approx\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\prod_{l=1}^{2}G(y_{il}-y_{jl},2\sigma_l^2)$$

$$\int\int f_{Y_1}(z_1)^2 f_{Y_2}(z_2)^2dz_1dz_2\approx\frac{1}{N^4}\prod_{l=1}^{2}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}G(y_{il}-y_{jl},2\sigma_l^2)\right)$$

$$\int\int f_{Y_1Y_2}(z_1,z_2)f_{Y_1}(z_1)f_{Y_2}(z_2)dz_1dz_2\approx \frac{1}{N^3}\sum_{i=1}^{N}\left[\prod_{l=1}^{2}\left(\sum_{j=1}^{N}G(y_{il}-y_{jl},2\sigma_l^2)\right)\right]$$

Now, the mutual information can be estimated directly from discrete set of data.

3.2 Information Theoretic Selection Procedure

Suppose we have two successfully trained neural networks (Net1 & Net2), which operate on the same data set. The networks are assumed to share the same structure. The fact each network has been successfully trained guarantees that the merged network can be also successfully trained with the same network structure. Let M denote the number of units in the hidden layer and restrict our discussion to MLP's that have only one hidden layer. We merge 2 MLP's into one by selecting M hidden units out of 2M units by information-based criterion. The criterion can be obtained by the following procedure.

The criterion to select a hidden unit is composed of 3 mutual informations (MI). The first one is MI between hidden unit' s output and the network' s target pattern. This value represents the usefulness of the hidden unit for the network. The second one is MI between hidden unit' s output and the target value of the other network. If this value is high then the unit is helpful to learn the patterns of the other network. The third one is the MI between hidden units. This measures the dependency of the unit. If a unit is highly dependent with the already selected units, the unit will be discarded.

Battiti [3] used mutual information to select input features. The selecting procedure proposed here is similar with his method. The main difference is the introduction of 'Cross Check' to addressing the information between networks. The procedure can be described by the following:

1) (Initialization) Set U {initial set of 2M hidden units}, S {empty set}
2) (Auto Check) Compute the MI between hidden units' outputs and the network' s target values (Net1 & Net1, Net2 & Net2): MI_A
3) (Cross Check) Compute the MI between hidden units' outputs and the other network' s target values (Net1 & Net2, Net2 & Net1): MI_B
4) (Choice of the first unit) Select the unit u that maximize MI_A+MI_B: set U $U-\{u\}$, S $S\cup\{u\}$
5) (Greedy Selection) Repeat until $|S|=k$:
   a) (Computation of the MI between units) For all couples (u,s) with $u\in U$, $s\in S$ compute MI $I(u;s)$, if it is not already available.
   b) (Selection of the unit) Choose unit u that maximizes $MI_A+MI_B-\beta\Sigma_{s\in S}I(u;s)$: set U $U-\{u\}$, S $S\cup\{u\}$

β is a parameter that regulates the relative importance of the MI between the candidate unit and the already-selected units. Bigger β means less dependent units are prefered. After selecting the hidden units, only the weights of the output layer are re-calculated linearly.

In this method, the dependency of a variable is measured by the summation of mutual informations between the variable and the selected variables. It cannot be guaranteed that the least dependent variable is selected, because generally, the mutual information with multiple variables is not expressed as the summation of each mutual information.

3.3 Merging by Error-based Criterion

The problem of selecting M units out of 2M units can also be solved by exhausted search. It requires $\binom{2M}{M}$ times of search. In general this is a too big number to be calculated. In this case, incremental selection method can also be applied. Because the linear learning process is very fast, the error of the network with selected node can be easily calculated. By selecting the unit that minimizes the error together with the already-selected units, the search iteration can be reduced to $(3M^2+M)/2$. The procedure is as following:

1) (Initialization) Set U {initial set of 2M hidden units}, S {empty set}
2) (Choice of the first unit) Select the unit u that minimizes the training or test error: set U $U-\{u\}$, S $S \cup \{u\}$
3) (Greedy Selection) Repeat until $|S|=k$:
   c) (Computation of the errors) For each units u in U, compute training or test error of the network with $S \cup \{u\}$ units.
   d) (Selection of the unit) Choose unit u that minimizes the error: set U $U-\{u\}$, S $S \cup \{u\}$

Because the unit is selected one after one, the searching space is much smaller than that of original problem. But this incremental selection does not guarantee the global solution to be found.

3.4 Merging of Two Networks for a New Task

Like the case of Bahrami [4]' s experiment, two networks can be merged to solve a new problem. The two networks are trained with data of their own. It is assumed that the information obtained from the previous learning is enough to solve the new problem. The procedure of merging is similar to that of section 3.2 except that the target pattern used here is the data of the new problem.

Although the sizes of the networks to be merged are the same, we cannot say that the equally-structured nework can perfectly learn the new task. It can be necessary to select more units than the network to be merged, depending on the complexity of the new task.

4. Experimental Results

5-bit-parity problem is tested in the experiments. Input is binary value of length 5 and the desired output is whether the number of 1 in the input string is odd or even. For training efficiency, input and output patterns are converted from {0,1} {-1,1}. Network structure of 5-4-1 is used for both MLP' s. 4 hidden layer units are selected out of 8 units. Backpropagation with moment and adaptive learning rate is used for training algorithm. As the selection criteria, quadratic mutual information is used.

Sharing Common Data Set: In this case, two networks are trained with the whole data set. Table 1 shows MSE error' s mean of 10 simulations. At first the merged network' s error is slightly bigger than those of the source networks. But after a little amount of training, the error of the merged network becomes smaller than those of the trained source networks. This can be interpreted as merging results in weight perturbation.

network	MSE	MSE after 500 interations of training
net 1	0.186	0.173
net 2	0.266	0.257
merged net	0.268	0.161

Table 1. Result of network merging (common data set)

Not Sharing Common Data Set: In this experiment, two networks trained with their own data set are merged. After merging process, all three networks are trained with the whole data set for 500 epochs. Table 2 shows the mean of 10 simulation results. The fact that merged network' s error is bigger, means two source networks don' t have enough information to solve the whole problem. In this case, the use of network merging may deteriorate the performance.

network	MSE	MSE after 500 interations of training
net 1	0.077	0.273
net 2	0.036	0.217
merged net	0.838	0.294

Table 2. Result of network merging (unique data set)

5. Concluding Remarks

Merging two networks enables us to get a smaller network than sum of original ones. By utilizing the information of the already trained networks, the learning process can be done very quickly. It is also beneficial that we can preserve the networks' structure and don't have to import a new topology.

By experiment it is clear that only when the source networks have enough information, we can expect the merged one to show desired performance.

References

[1] S. Haykim, *Neural Networks: A Comprehensive Foundation*. 2nd Ed., Prentice-Hall, 1999

[2] J. W. Guan and D. A. Bell, "Evidence Theory and its Application," vol.1, North-Holland, 1991

[3] R. Battiti, "Using Mutual Information for Selecting Features in Supervised Neural Net Learning," *IEEE Trans. Neural Networks*, vol. 5 no. 4, pp.537-550, 1994

[4] M. Bahrami, "Integration of Knowledge Acquired by Different Neural Networks," *First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, 1993

[5] J. T. Lo and D. Bassu, "Adaptive Multilayer Perceptrons With Long- and Short-Term Memories," *IEEE Trans. Neural Networks*, vol. 13 no. 1, pp.22-33, 2002

[6] P. Burrascano and D. Pirollo, "Merging Information in the Data and Weight Spaces," 8th Mediterranean Electrotechnical Conference, vol.2, pp.617-620, 1996

[7] J.C. Principe, J.W. Fisher III, and D. Xu, "Information Theoretic Learning," in *Unsupervised Adaptive Filtering*, Simon Hakin, Ed. Wiley, New Youk, NY, 2000

[8] E. Parzen, "On the estimation of a probability density function and the mode", Ann. Math. Stat. 33, p.1065, 1962