Enhanced RBF Network by Using ART2 Algorithm and Fuzzy Control Method

Kwang-Baek KIM\textsuperscript{(2)}, Member, Sung-Kwan JE\textsuperscript{††}, and Young-Ju KIM\textsuperscript{†}, Nonmembers

1. Introduction

Many studies on improving the learning time and the generalization ability of the learning algorithm of neural network have been performed. As a result, RBF (Radial Basis Function), which has been used for multivariate analysis and interpolation of statistics, was used for organizing the neural network model by Broomhead and Low for the first time. Then RBF network was proposed by Watanabe et al.\cite{1}. RBF network has the characteristics of short learning time, generalization and simplification etc., applying to the classification of learning data and the nonlinear system modeling.

The RBF network is a feed-forward neural network that consists of three layers, input layer, middle layer and output layer. In the RBF network, because the operations required between layers are different, learning algorithms between layers can be mutually different. So, the optimum organization between layers can be separately constructed\cite{2}. Approaches to the composition of layers in the RBF network are classified to three types: The first type is the ‘fixed centers selected at random’ which selects nodes of the middle layer randomly from the learning data set. The second is the ‘self-organized selection of centers’ which decides the middle layer according to the form of self-organization and applies the supervised learning to the output layer. The last one is the ‘supervised selection of centers’ which uses the supervised learning for the middle layer and the output layer. The middle layer of the RBF network executes the clustering operation, classifying input data set to homogeneous clusters. The measurement of homogeneity in clusters is the distance between vectors in clusters. And the classification of input data to a cluster means that the distances between input data and each vector in the cluster are shorter than or equal to the fixed radius. But, the use of a fixed radius in clustering causes wrong classifications. Therefore the selection of the organization for middle layer determines the overall efficiency of the RBF network\cite{3}. Therefore this paper proposes and evaluates the enhanced RBF network that uses ART2 to organize the middle layer efficiently and applies the auto-turning method of adjusting learning rate and momentum using the fuzzy control system for the arbitration of the connected weight between middle layer and output layer.

2. Related Studies

2.1 Delta-Bar-Delta Algorithm

Delta-bar-delta algorithm\cite{4}, which improved the quality of backpropagation algorithm, enhances learning quality by arbitrating learning rates dynamically for individual connected weights by means of making delta and delta-bar. When the multilayer neural network consists of the input layer $i$, the middle layer $j$ and the output layer $k$, the delta-bar-delta algorithm shows the change of connected weights like Eq. (1).

\begin{equation}
\Delta_{kj}(t+1) = \Delta_{kj}(t) - \alpha_{kj}(t+1) \frac{\partial E}{\partial w_{kj}}
\end{equation}

\begin{equation}
\Delta_{kj} = w_{kj}(t) - \alpha_{kj}(t+1)\delta_{kj} z_j
\end{equation}

where $\alpha_{kj}$, $\delta_{kj}$ and $z_j$ indicate the learning rate, the error signal of the output layer and the output of a neuron of the middle layer, respectively. The delta for each neuron of the middle layer and the output layer is made by means of Eq. (2) and Eq. (3), respectively.

\begin{equation}
\Delta_{ji} = \frac{\partial E}{\partial w_{ji}} = -\delta_{ji} x_i
\end{equation}

\begin{equation}
\Delta_{kj} = \frac{\partial E}{\partial w_{kj}} = -\delta_{kj} z_j
\end{equation}
The formulas of making delta-bar are like Eq. (4) and Eq. (5), and the delta-bar-delta rule uses the combination of the current information of each output neuron and the past information derived by Eq. (4) and Eq. (5).

\[
\Delta_{ji}(t) = (1 - \beta) \cdot \Delta_{ji}(t) + \beta \cdot \Delta_{ji}(t - 1) \quad (4)
\]

\[
\Delta_{kj}(t) = (1 - \beta) \cdot \Delta_{kj}(t) + \beta \cdot \Delta_{kj}(t - 1) \quad (5)
\]

The value of parameter \( \beta \) in Eq. (4) is the fixed constant between 0 and 1.0. The variation of learning rate in terms of the change direction of delta and delta-bar is as follows: If the connected weight changes to the same direction in the successive learning process, the learning rate will increase. At this point \( \Delta_{kj}(t - 1) \) and \( \Delta_{kj}(t) \) has the same sign. On the other hand, if the signs of \( \Delta_{kj}(t - 1) \) and \( \Delta_{kj}(t) \) are different, the learning rate will decrease as much as the ratio of \( 1 - \gamma \) of the present value. Therefore the variation of learning rate for each layer can be expressed like the following formulas:

\[
\alpha_{ji}(t + 1) = \begin{cases} 
\alpha_{ji}(t) + k & : \Delta_{ji}(t - 1) \cdot \Delta_{ji}(t) > 0 \\
(1 - \gamma) \cdot \alpha_{ji}(t) & : \Delta_{ji}(t - 1) \cdot \Delta_{ji}(t) < 0
\end{cases} \quad (6)
\]

\[
\alpha_{kj}(t + 1) = \begin{cases} 
\alpha_{kj}(t) + k & : \Delta_{kj}(t - 1) \cdot \Delta_{kj}(t) > 0 \\
(1 - \gamma) \cdot \alpha_{kj}(t) & : \Delta_{kj}(t - 1) \cdot \Delta_{kj}(t) < 0
\end{cases} \quad (7)
\]

2.2 ART2-Based RBF Network

The learning of ART2-based RBF network is divided into two stages. In the first stage, competitive learning is applied as the learning structure between input layer and middle layer. And the supervised learning is accomplished between middle layer and output layer [5], [6]. Output vector of the middle layer in the ART2-based RBF network is calculated by Eq. (8), and as shown in Eq. (9), the node having the minimum output vector becomes the winner node.

\[
O_j = \frac{1}{N} \sum_{i=1}^{N} \left( |x_i - w_{ji}(t)| \right) \quad (8)
\]

\[
O_j^* = \min(O_j) \quad (9)
\]

where \( w_{ji}(t) \) is the connected weight value between input layer and middle layer.

In the ART2-based RBF network, the node having the minimum difference between input vector and output vector of the hidden layer is selected as the winner node of the middle layer, and the similarity test for the winner node selected is the same as Eq. (10).

\[
O_j^* < \rho \quad (10)
\]

where \( \rho \) is the vigilance parameter in the formula.

The input pattern is classified to the same pattern if the output vector is smaller than the vigilance parameter, and otherwise, to the different pattern. The connected weight is adjusted to reflect the homogeneous characteristics of input pattern on the weight when it is classified to the same pattern. The adjustment of the connected weight in ART2 algorithm is as follows:

\[
w_{jk}(t + 1) = w_{jk}(t) \cdot u_n + \frac{x_i}{u_n + 1} \quad (11)
\]

where \( u_n \) indicates the number of updated patterns in the selected cluster. The output vector of the middle layer is normalized by Eq. (12) and applied to the output layer as the input vector.

\[
z_i = 1 - \frac{O_j}{N} \quad (12)
\]

The output vector of the output layer is calculated by Eq. (13).

\[
O_k = f \left( \sum_{j=1}^{M} w_{kj} \cdot z_j \right) \quad (13)
\]

\[
f(x) = \frac{1}{1 + e^{-x}} \quad (14)
\]

The error value is calculated by comparing the output vector with the target vector. The connected weight is adjusted like Eq. (16) using the error value.

\[
\delta_k = (T_k - O_k) \cdot O_k \cdot (1 - O_k) \quad (15)
\]

\[
w_{kj}(t + 1) = w_{kj}(t) + \alpha \cdot \delta_k \cdot z_j \quad (16)
\]

3. Enhanced RBF Network

The learning rate used in the RBF network is the parameter adjusting the variation of connected weights. It means that the learning rate is used to prevent the vibration of connected weights having no the right local minimum. And the fixing of learning rate has an influence upon the time of learning and the convergence degree of learning. Generally, it is efficient in improving the performance of learning that the learning rate is fixed to the large value in the case that the change of TSSE (Total Sum of Squares of Error) is very little, and oppositely to the small value in the case that connected weights arrive the local minimum.

The enhanced RBF network applies ART2 to the learning structure between the input layer and the middle layer and proposes the auto-tuning method of arbitrating the learning rate for the adjustment of the connected weight between the middle layer and the output layer.

The traditional RBF network continues learning until the TSSE becomes less than the error criteria. But the error of output values of some nodes in the output layer may not continue falling off, so that it may require much time for learning and cause the incorrect classification of features of input patterns and a decrease in the success rate of recognition. Therefore, in the proposed enhanced RBF network, when the absolute value of the difference between the output vector and the target vector for each pattern is below 0.1, it
is classified to the accuracy, and otherwise to the inaccuracy. And the learning rate and the momentum are arbitrated dynamically by applying the numbers of the accuracy and the inaccuracy to the input of the fuzzy control system.

Generally, the learning process is divided to the three stages: the error convergence stage when the error is rapidly decreased at the beginning of learning, the competition stage when the error does not change hardly because of little variations of connected weights induced by the conflict between the change of connected weight of the particular pattern and the one of the different pattern, and the superiority stage when the error is rapidly decreased again according to the unexpected learning of patterns not being learned. Among the three stages, the competition stage takes up the great part of learning time relatively [8].

In this paper, for the reduction of time required in the competition stage, the method was proposed that arbitrates dynamically the learning rate by applying the numbers of the accuracy and the inaccuracy to the input of the fuzzy control system. The numbers of the accuracy and the inaccuracy are used to distinguish the competition stage in the learning process. The learning process showing the small number of the accuracy and the large number of the inaccuracy can be defined as the error convergence stage, and when the number of the accuracy is large and only increasing, the process is defined to be in the superiority stage. On the contrary, when the numbers of the accuracy and the inaccuracy show the variations with changing direction or little variations, the process can be defined as the competition stage. Figure 1 shows the membership function to which the accuracy belongs, whereas Fig. 2 shows the membership function to which the inaccuracy belongs.

In the Boltzmann training method, each soma consists of two types of nodes, the visible nodes and the hidden nodes, and the visible nodes are classified to the input nodes and the output nodes. Ackley [9] proposed that the number of hidden nodes required in the Boltzmann training must be above \( \log_2(\text{the number of visible nodes}) \). In this paper, the experiment using the Ackley’s formula showed that the numbers of nodes of the input layer and patterns have a sensitive influence upon the performance of learning. Therefore, the numbers are applied to determine the intervals of the proposed membership functions. The values of \( C_{\text{low}} \) and \( C_{\text{high}} \) are calculated by formula (17) and (18).

\[
C_{\text{low}} = \log_2(N_i + N_p) \\
\{ N_i : \text{the number of input nodes} \\
\{ N_p : \text{the number of patterns} \} \tag{17}
\]

\[
C_{\text{high}} = C_{\text{low}} - C_{\text{low}} \tag{18}
\]

In Figs. 1 and 2, F, A and T are the membership functions indicating false, average and true, respectively. When the rule of controlling fuzzy to arbitrate the learning rate is expressed with the form of if \( \sim \) then, it is as follows:

\[
R_1: \text{if correct is } F, \text{ incorrect } F \text{ Then } \alpha \text{ is } B \\
R_2: \text{if correct is } F, \text{ incorrect } A \text{ Then } \alpha \text{ is } B \\
R_3: \text{if correct is } F, \text{ incorrect } T \text{ Then } \alpha \text{ is } B \\
R_4: \text{if correct is } A, \text{ incorrect } F \text{ Then } \alpha \text{ is } M \\
R_5: \text{if correct is } A, \text{ incorrect } A \text{ Then } \alpha \text{ is } M \\
R_6: \text{if correct is } A, \text{ incorrect } T \text{ Then } \alpha \text{ is } M \\
R_7: \text{if correct is } T, \text{ incorrect } F \text{ Then } \alpha \text{ is } S \\
R_8: \text{if correct is } T, \text{ incorrect } A \text{ Then } \alpha \text{ is } S \\
R_9: \text{if correct is } T, \text{ incorrect } T \text{ Then } \alpha \text{ is } S \\n\]

Figure 3 shows the output membership function calculating the learning rate, which is going to be applied to learning. In Fig. 3, S, M and B are the membership functions indicating small, medium and big, respectively. When accuracy and inaccuracy are decided as the input value of the fuzzy control system, membership degrees of accuracy and inaccuracy for each membership function are calculated. After the calculation of membership degree for each membership function, the rule of fuzzy control is applied and the inference is accomplished by means of Max_Min method. The learning
rate, which is going to be used for learning, is calculated by defuzzifier method after the fuzzy inference. Formula (19) shows the center of gravity, which is used for the defuzzification [7].

\[
\alpha = -\frac{\sum \mu(y) \cdot y}{\sum y} \tag{19}
\]

Momentum can be used to determine the next value of connected weight by reflecting properly the previous values of connected weight. In this paper, momentum is adjusted based on the proposed method arbitrating dynamically the learning rate. That is, when the connected weight is close to the local minimum and the learning rate becomes small, momentum is fixed to a large value for the speedup of learning. Therefore, momentum is calculated by formula (20).

\[
\mu = \zeta - \alpha \tag{20}
\]

where \(\zeta\) is the parameter between 1.0 and 1.5, which is given empirically.

4. Experiments and Performance Evaluation

We implemented the enhanced RBF network proposed with C++ Builder 6.0 and executed the experiment for performance evaluation on IBM compatible PC in which Intel Pentium-IV CPU and 256 MB RAM were mounted. We analyzed the number of epoch and the convergence by applying 136’s number patterns having 10 \times 10 in size, which are extracted from the citizen registration cards, to the conventional delta-bar-delta method, the ART2-based RBF network and the learning algorithm proposed in this paper. Figure 4 shows patterns of numbers which were used for learning. Table 1 shows target vectors. Table 2 shows parameters of each algorithm which were used for the experiment and Table 3 shows the result of learning. In Table 2, \(\alpha\) indicates the learning rate, \(\rho\) the vigilance parameter of ART2, \(\zeta\) the parameter for calculation of momentum. And \(\beta, \kappa\) and \(\gamma\) indicate the parameters fixed by the delta-bar-delta algorithm.

The experiment have been executed 10 times under the criterion of classifying input pattern to the accuracy when the absolute value of the difference between the output vector of input pattern and the target vector is below \(\epsilon (\epsilon \leq 0.1)\) in 10000’s epoch executions. In the experiment for delta-bar-delta algorithm, when the number of nodes of the middle layer was fixed to a value between 5 and 10, the experiment with 10’s nodes showed the least time for learning. So, in Table 2, the result for delta-bar-delta algorithm was derived when the number of nodes of the middle layer was fixed to 10. In the ART2-based RBF network and the proposed method, when the vigilance parameter used to create and update nodes of the middle layer was fixed to 0.1, 10’s nodes were created in the middle layer.

The fact that the proposed method is more enhanced than conventional methods in terms of learning speed and convergence is verified in Table 3. Moreover the proposed method did not react sensitively to the number of iterative learning and the convergence, whereas conventional methods did. Consequently, the insensibility of the proposed method enhances the efficiency of learning. As showed in Table 3, delta-bar-delta algorithm needs the more computation time for the adjustment of learning rate than the proposed method.

Figure 5 shows the graph for the change rate of TSS (Total Sum of Square) of error according to the number of epoch. As shown in Fig. 5, the proposed method has faster speed of primary convergence and smaller TSS of error than conventional methods.

Through experimental results, we found that the proposed method spent less time for training compared with the conventional training method, and had good convergence ability.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Comparision of convergence among each algorithm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning method</td>
<td># of experiment</td>
</tr>
<tr>
<td>Delta-bar-delta</td>
<td>10</td>
</tr>
<tr>
<td>ART2-based RBF network</td>
<td>10</td>
</tr>
<tr>
<td>Proposed method</td>
<td>10</td>
</tr>
</tbody>
</table>

Fig. 5 Graph of total sum of squares.
5. Conclusions

The learning of the ART2-based RBF network is divided to two stages. At the first stage the competitive learning is performed between the input layer and the middle layer, and at the second stage the supervised learning is performed between the middle layer and the output layer. In this paper, an enhanced RBF network is proposed, which uses ART2 algorithm between the input layer and the middle layer to enhance the efficiency of learning of conventional ART2-based RBF network and, to adjust the weight value efficiently between the middle layer and the output layer, applies the auto-tuning method of arbitrating the learning rate and the momentum automatically by means of the fuzzy control system. In the proposed auto-tuning method of the learning rate and the momentum, when the absolute value of the difference between the output vector and the target vector for the input pattern is equal to or below than $\varepsilon$, the input pattern is classified to the accuracy, and otherwise to the inaccuracy. Then, applying the numbers of the accuracy and the inaccuracy to the fuzzy control system, the learning rate is arbitrated dynamically. The momentum is arbitrated dynamically by using the adjusted learning rate, so that the efficiency of learning is improved.

The experiments of applying the proposed method to the classification of number patterns extracted from the citizen registration card shows 2 results related to performance: First, the proposed method did not react sensitively to the number of learning and the convergence, whereas conventional methods did, and second, the total sum of square has decreased remarkably than conventional methods. Therefore, the efficiency of learning in the proposed method is concluded to be enhanced. The study on the method generating the optimized middle layer by enhancing the efficiency of ART2 algorithm will be the subject of study in the future.

References