Simultaneously Structural Learning and Training of Neurofuzzy GMDH using GA

A. Sharifi*, M. Teshnehlab**
* Computer Department of Islamic Azad university Science and Research Branch of Tehran, Iran
** K. N. Toosi university of technology, Tehran, Iran

Abstract—This article presents a new approach for Structural Learning of Neurofuzzy (NF-) GMDH networks, based on Genetic Algorithm (GA) optimization. Conventional methods, prune unnecessary links and units from the large network by minimizing the derivatives of the partial description. In proposed method pruning of needless links, units and fuzzy rules in each partial description, has been done by adding some extra binary weights to the conclusion part of each partial description. Two kinds of GA also proposed, necessary fuzzy rules in the conclusion part of each partial descriptions in NF-GMDH network, are chosen by using the binary-coded GA, and system parameters are adjusted by using the real-coded GA. Finally the newly proposed method is validated in classification of Iris data.

Keywords—GA algorithm, RBF networks, GMDH networks, Neurofuzzy and Pruning

I. INTRODUCTION

As one of neural networks (NNs), Group Method Data Handling (GMDH) [1] model is one of the network has been proposed, and employed for prediction, modeling and control of the nonlinear systems and also in data classification.

NF-GMDH [2], [3] model is one of the network type GMDH whose partial descriptions are represented by Radial Basis Function (RBF) network [4], [5]. The RBF network is a technique for interpolating data in multidimensional spaces. The networks have the architecture that uses a single internal layer of locally tuned processing units and are called 'localized receptive fields' [4]. Brown and Harris demonstrated in their book entitled 'Neurofuzzy adaptive modeling and control' [6] that there exist an invertible relationship between fuzzy logic systems and RBF networks, with each inheriting the properties of the other. For training this type of network, conventional backpropagation (BP) learning with some accelerated methods can be adopted and it is shown that it can converge as fast as the NNs.

The NF-GMDH network is a kind of networks with a structure which is configured through training sequence, but there is a problem of determining the optimal structure of these kinds of networks. There are several methods for solving this problem and they can categorize into two groups [7], [15]: constructive methods and destructive methods. For first approach we start with a small structure of network and expand its structure through training sequence, but for second approach we start with a large structure of network and try to reduce and optimize its structure through training. The proposed method uses destructive approach.

Previous works such as Structural Learning with Forgetting (SLF) [8], [9] or M-Apoptosis SLF (MSLF) [10] prune unnecessary links and units from large network by minimizing the derivatives of the partial description. So training sequence consists of two steps, first step is obtaining optimal structure of network, and at second step, train minimized network with conventional BP method.

In this paper, we proposed a new method for pruning needless units, links and fuzzy rules of the conclusion parts of partial descriptions in the NF-GMDH networks, based on GA optimization [11]. This newly proposed algorithm can use simultaneously for structural identification and training network parameters, and therefore extra training phase, such as BP used in SLF and MSLF, is not needed. The basic idea of this new method is based on adding some extra binary weights to the conclusion parts of the partial descriptions, and these extra binary weights are constructing the structure of network through training sequence. For this purpose we introduce two kinds of GA. In constructive the network structure we use binary-coded GA and for determining system parameters we use real-coded GA. Finally, the effectiveness of the proposed method is validated in the classification of Iris data.

II. NEUROFUZZY GMDH SCHEME

A. Outline of NF-GMDH Network

In the perceptron type GMDH algorithm, all partial descriptions with one or two variables chosen from all the input variables in each layer are evaluated, and several of them are selected. The network type NF-GMDH have been proposed, whose partial descriptions are represented by the RBF networks. In the net, two input variables are introduced in each partial description. Let the number of partial descriptions in each layer be M and the number of layers be P. The fuzzy partial description in this paper is as follows:

Let \( A_{ik}(x_i) \) denote the membership function of the \( k-th \) fuzzy rule in the domain of \( i-th \) input variable. The compatibility degree of the premise part of the \( k-th \) fuzzy rule for the \( m-th \) partial description in the \( p-th \) layer is computed with the algebraic product operation of the membership functions as:

\[
\text{compatibility degree} = \prod_{i=1}^{n} A_{ik}(x_i)
\]
\[ \mu_{im}^{pm} = \prod_{i=1}^{L} A_{ui}^{pm}(x_i) \]  

(1)

Where, \( L \) can be 1 or 2 for the partial descriptions of the NF-GMDH. The conclusion part of the fuzzy inference rule which infers \( y \) is simplified as a real number \( w_j \). For the \( m\text{-th} \) partial description in the \( p\text{-th} \) layer the output is written as:

\[ y^{pm} = \sum_{k=1}^{K} \mu_k^{pm} w_k^{pm} \]  

(2)

This model is called the simplified fuzzy reasoning when following membership function is chosen.

\[ A_{ui}^{pm}(x_i) = \exp \left\{ - \frac{(x_i^{pm} - a_{ui}^{pm})^2}{b_{ui}^{pm}} \right\} \]  

(3)

Equation (2) can be regarded as the RBF network which can be capable of approximating any continuous mapping within an arbitrary accuracy. The final output of NF-GMDG, \( y \), is the average of outputs in the last layer.

\[ y = \frac{1}{M} \sum_{m=1}^{M} y^{pm} \]  

(4)

Fig. 1 shows the model structure of three layered NF-GMDH which has three partial descriptions per layer. By consideration \( \hat{y} \) as target value, the performance index of the error is calculated as:

\[ E = \frac{1}{2} (y - \hat{y})^2 \]  

(5)

The newly proposed algorithm successively proceeded to minimizing \( E \) by using the GA optimization.

III. PRUNING ALGORITHM BASED ON GA

A. Problems of conventional methods

The conventional BP algorithm has a vital problem. The NF-GMDH is constructed by suitable combination of partial descriptions and links. The problem is that conventional BP algorithm can not find which link or partial description needs to be eliminated, therefore network structure has some needless links and partial descriptions. The SLF algorithm has two kinds of problems. One is that it needs extra training phase for adjusting parameters of minimized network structure. Another problem is finding a threshold value for the deletion needless links and partial descriptions. The MSLF [10] algorithm needs extra training phase as well and also it needs training phase with SLF [8], [9].

The newly proposed method can adjust the structure of the network and parameters, using the GA [11]. In constructing the network structure, a selection coefficient "0", "1" is assigned for each weight in conclusion part of partial description and the fuzzy rule is given for the partial description whose selection coefficient is equal to "1". For structural learning, simple binary GA is adopted and for system parameter, learning a real-coded GA is employed.

B. GA for designing optimum structure

In this scheme, for each weight in conclusion part of the partial descriptions, extra binary weight is assigned. That is the coefficient "0" means the needless fuzzy rule and the coefficient "1" is the opposite one. The procedure of the GA for finding the structure is as follows:

Initialization Depend on chosen initial network structure, generate selection coefficients randomly. For example, if structure of network is same as network shown in Fig. 1, and there are 4 fuzzy rules in each partial description, generate 36 binary selection weights. 12 bits for conclusion parts of layer \( P_1 \), 12 bits for layer \( P_2 \) and last 12 bits for layer \( P_3 \). Here \( \text{Pop}_ \text{Max} \) represented the number of binary weights that generated.

Handling Structure Because generation of structural weights is randomly, some illegal structure maybe generated. In this case we chose two strategies in order to correct this problem. Handling structure is done at each generation before computing fitness function. They are as follows:

Forward Chaining: In some situation both inputs of a partial description are disabled, but output link of this partial description is still remains active. Let \( P \) be the number of layers. In forward chaining strategy we start from layer 2 to layer \( P \) and eliminate all partial descriptions which both inputs of them in previous layer have been disabled. In this case elimination means changing coefficient "1" to "0" for rules in that partial description. Fig. 2 shows example of forward chaining.

Backward Chaining: After applying forward chaining some useless partial descriptions maybe occurred. In backward chaining strategy, in order to eliminate these useless partial descriptions we start from layer \( P - 1 \) to 1 and disable all fuzzy rules and output link of every partial description that its corresponding partial descriptions in next layer have been disabled. Fig. 3 shows example of backward chaining.
Crossover In designing structure part, we adopt one-point crossover [12], [16]. In each generation, the crossover point is given by random number. The crossover rate (denoted by \( P_c \)) is defined as a ratio of the number of offspring produced in each generation.

**Mutation** In this part, we use simple mutation method [12], [16]. In this method we select a single gene and toggle its value. The mutation rate (denoted by \( P_m \)) is defined as the percentage of the total number of genes that should be mutated.

**Selection** Enlarge sampling space is adopted. With this strategy, \( \mu \) parents and \( \lambda \) offspring compete for survival and the \( \mu \) best out of offspring and old parents are selected as parents of the next generation.

**Fitness function** This measurement explained next section. The measure that calculated by the suitable combination of selection coefficients and system parameters is set as the fitness value of each chromosome.

C. GA for designing system parameters

Here, a real-coded GA is employed in order to search system parameters for partial descriptions which are given a selection coefficient “1”. A search procedure of GA for designing system parameters is as follows:

**Initialization** Generate initial population by the real random number. Generate a set of parameters consist of \( a, b \) and \( w \) for each partial description. Here \( Pop\_Max \) represented the number of real parameter sets that generated.

**Crossover** Randomly chose two chromosomes \( A \) and \( B \) from the parent population. Two types of crossover are proposed and in each generation one of them chosen randomly to perform crossover operation. The crossover rate (denoted by \( P_c \)) is defined as a ratio of the number of offspring produced in each generation. These two types are convex and linear crossover and are as follows [12], [16]:

\[
A' = \lambda_1 A + \lambda_2 B \\
B' = \lambda_1 B + \lambda_2 A
\]

Convex Crossover

\[
\lambda_1 + \lambda_2 = 1, \quad \lambda_1 > 0 \quad \lambda_2 > 0
\]

Linear Crossover

\[
\lambda_1 + \lambda_2 = 1, \quad \lambda_1 > 0 \quad \lambda_2 > 0
\]

**Mutation** Here, we use uniform mutation method. This one simply replaces a gene (real number) with a randomly selected real number within a specified range. The mutation rate (denoted by \( P_m \)) is defined as the percentage of the total number of genes that should be mutated [12], [16].

**Selection** Here, enlarge sampling space is employed as well.

**Fitness function** The fitness function can be freely designed. In this paper, the following equation is considered:

\[
Eval(p) = \frac{1}{N} \sum_{i=1}^{N} (y(i) - \hat{y}(i))^2
\]

Where \( y(i) \) and \( \hat{y}(i) \) denote the target value and the final output of network and \( N \) denotes the number of data.

IV. CLASSIFICATION OF IRIS DATA

In this section one pattern recognition problem is used to show the performance of the newly proposed method. The performance of a classifier is measured by its error rate. The Iris data is the standard discriminate analysis example for benchmark test of classification methods [13]. It contains 50 measurements of four features from each three species **Setosa**, **Versicolor** and **Virginica**. Features are sepal length, sepal width, and petal length and petal width. The data set consist of 150 cases, 50 for each class. The characteristic of the data are shown in Fig. 4. In the figure \( \Delta \), \( O \), \( \Box \) denote Versicolor, Setosa and Virginica, respectively.

According to the classification results in Ohtani et al. [3], we employed the network structure shown in Fig. 5 with \( K=4 \), by which comparatively lower error rate than other methods [14] is obtained. The target on Setosa, Versicolor and Virginica were set to 0.0, 0.5 and 1.0 respectively. We classified according to followings rules: Setosa if \( y<0.33 \), Virginica if \( y\geq0.67 \) else Versicolor. 120 cases (40 cases for each class) were used as training data, and the other 30 cases as checking data.

Although each building block of NF-GMDH is a function with two input variables, both variable do not necessarily correlate with its output. Fig 6 shows an example of the building block, in which the input \( x_1 \) is irrelevant to output, i.e. \( \frac{\partial \hat{y}}{\partial x_2} \approx 0 \). In this example, \( x_2 \) can be eliminated.
This simulation study is shows that GA can clarify the structure of Iris data and can improve the number of parameters. The epochs of conventional BP learning, SLF and MSLF were 10000, 5000 and 5000 respectively, where MSLF adapted to the result after learning by SLF. The population number, generation times, $P_c$ and $P_m$ of GA were 10, 40, 0.5 and 0.1, respectively. Table I shows the resultant averaged $|\partial y^m / \partial x_i^m|$ for classification of Iris data with BP, SLF and MSLF methods.

Table II shows the structure coefficient of fuzzy rules that obtained by GA method. Fig. 7 shows the network structure based on coefficients in Table II in which the unnecessary fuzzy rules have been deleted.

Table III shows the results where $E(T)$ and $E(C)$ mean discrimination error rate for training and checking, respectively. The 10-fold cross validation is used to comparison of the classification ability, of the newly proposed algorithm and conventional BP. The averaged error rates for training and checking set of the network shown in Fig. 5 were 0.013 and 0.033, respectively and the averaged error rates for training and checking set for the pruned network shown in Fig. 7 were 0.018 and 0.037, respectively. There is no much loss of generalization ability after elimination of 85 parameters out of 160.
TABLE III

<table>
<thead>
<tr>
<th></th>
<th>BP</th>
<th>SLF</th>
<th>MSLF</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E(T)$</td>
<td>0.83</td>
<td>3.12</td>
<td>2.50</td>
<td>1.04</td>
</tr>
<tr>
<td>$E(C)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Number of Parameters</td>
<td>160</td>
<td>124</td>
<td>124</td>
<td>75</td>
</tr>
</tbody>
</table>

Fig. 7 Network structure for Iris data after classification

V. CONCLUSION

In this paper a structural learning method for NF-GMDH has been proposed. According to the proposed method, redundant fuzzy rules, links and units have been deleted from the network structure. Also two kinds of GA have been proposed. The binary-coded GA selects the necessary fuzzy rules in each partial description. The real-coded GA calculates the parameters values for the fuzzy rules which are remained in reconstructing the partial description. In numerical example we show the network structure is clarified by using this GA method. The future work includes improvement the convergence speed of this newly proposed scheme.

REFERENCES