Incremental Leaning Algorithm for Self-organizing Fuzzy Neural Network

Xionghui Long
Guangzhou Institute of Railway Technology
Guangzhou, China
longxh@126.com

Dan Su
Guangzhou Institute of Railway Technology
Guangzhou, China
sudan@126.com

Rong Hu
School of Computer Science and Technology,
Nanjing University of Science and Technology,
Nanjing, China

Abstract—This paper proposed an incremental learning algorithm for self-organizing fuzzy neural networks (ILSFNN) based on extended radial basis function neural networks, which are functionally equivalent to Takagi–Sugeno–Kang fuzzy systems, is proposed. First, a self-organizing clustering approach is used to establish the structure of the network and obtain the initial values of its parameters. Then, a hierarchical on-line self-organizing learning paradigm is employed so that not only parameters can be adjusted, but also the determination of structure can be self-adaptive without partitioning the input space a priori. Simulation studies and comprehensive comparisons with some other learning algorithms demonstrate that the proposed algorithm is superior in terms of simplicity of structure, learning efficiency and performance.

Index Terms—Incremental learning; Fuzzy neural networks; TSK fuzzy reasoning; Self-organizing

I. INTRODUCTION

The complexity and the dynamics of many real-world problems require sophisticated methods and tools for building intelligent information systems (IS). Such systems should be able to learn and deal with different types of data and knowledge through interaction with the environment in an incremental way.

Recently, neuro-fuzzy systems, as a combination of neural networks and fuzzy logic, have attracted much attention because they provide both the merits of the learning ability of neural networks and the knowledge representation of fuzzy systems [1,2]. Fuzzy neural networks (FNNs) are hybrid systems that combine the theories of fuzzy logic and neural networks, thus can make effective use of easy interpretability of fuzzy logic, as well as superior learning ability and adaptive capability of neural networks.

Although some special FNNs (fuzzy neurons and fuzzy weights) have been presented, the typical approach of designing FNNs is to build neural networks which are designed to approximate a fuzzy algorithm or a process of fuzzy inference through the structure of neural networks [3-4,7-8]. Incremental learning is concerned with learning data as the system operates (usually in real time) and data might exist only for a short time. FNN models for incremental learning are introduced and studied in [9,13]. Nevertheless, in most FNNs, structure identification is still time consuming. In order to cope with this problem, we presented an incremental learning algorithm for self-organizing fuzzy neural networks (ILSFNN) based on extended radial basis function (RBF) neural networks, which are functionally equivalent to TSK fuzzy systems [5], has been developed in [6]. The methodology therein is summarized as follows: the system starts with no rules. Fuzzy rules can be recruited or deleted dynamically according to an input-output data set. This algorithm has fast learning speed because structure and parameter identification are done automatically and simultaneously without partitioning the input space and selecting initial parameters a priority.

This paper is organized as follows: Section 2 describes the architecture of the ILSFNN. Based on the proposed architecture, a novel learning algorithm is given in detail in Section 3. Section 4 presents the simulation results and some comparative studies with other learning algorithms. Finally, conclusions are drawn in Section 5.

II. ARCHITECTURE OF THE ILSFNN

The architecture of the proposed ILSFNN is a five-layer FNN shown in Figure 1. Without loss of generality, we consider a multi-input single-output (MISO) fuzzy model:

\[ R_k : \text{IF} \ x_1 \ \text{is} \ A_1^k \ \text{AND} \ ... \ \text{AND} \ x_n \ \text{is} \ A_n^k \ \text{THEN} \ y_k = w_k \]

where \( R_k \) is the \( k \) th rule (\( 1 \leq k \leq L \)), \( x = \{ x_i \}_{i=1}^n \) is input linguistic vector, \( w_k \) is the \( k \) th consequent parameter. Fuzzy sets are defined by Gaussian membership functions:

\[ \mu_k(x) = \exp\left(-\left(x - c_k\right)^2/\sigma_k^2\right), \]

\( i=1,2,..,n, k=1,2,..,L \) \hspace{1cm} (I)
Figure 1 Architecture of ILSFNN

where $\mu_k$ is the $k$th membership function of $x_i$, $c_k$ and $\sigma_k$ are the center and width of the Gaussian function, respectively. $n$ is the number of input variables and $L$ is the number of membership functions for each input variable.

Layer 1: Input layer. Each node in layer 1 represents an input linguistic variable.

Layer 2: Each node in layer 2 represents a membership function (MF) which is in the form of Gaussian functions.

Layer 3: Fuzzy Inference layer. Product inference method is used here, the number of nodes in layer 3 is equal to the number of fuzzy rules. Each node computes the rule activation strength, for the $k$th rule $R_k$, its output is

$$
\mu_k(x) = \prod_{i} \mu_{i_k}(x_i), \quad k = 1, 2, \ldots, L \quad (2)
$$

Layer 4: Normalized layer. The number of $N$ (normalized) nodes is equal to that of $R$ nodes. The output of $N_k$ is

$$
\psi_k(x) = \frac{\mu_k(x)}{\sum_{i=1}^{n} \mu_i(x)}, \quad k = 1, 2, \ldots, L \quad (3)
$$

Layer 5: Output layer. The single node computes the output value $y$:

$$
y = \sum_{k=1}^{K} \psi_k(x) w_k, \quad (4)
$$

where $w_k$ is the consequent parameter or connection weight of the $k$th rule.

III. AN INCREMENTAL LEARNING ALGORITHM OF THE ILSFNN

If the number of fuzzy rules needs to be identified, we cannot choose the ILSFNN structure a priority. This leads us to develop a new type of learning algorithm for the ILSFNN which is capable of automatically determining the number of fuzzy rules for the desired system’s performance.

A. Criteria of rules generation

System errors: The error criterion can be described as follows: For the $i$th observation $(x_i; y_p)$, where $x_i$ is the input vector and $y_p$ is the desired output, compute the overall ILSFNN output $\hat{y}_p$ on the existing structure according to Eq. (4).

Define

$$
E = \sum_{p=1}^{p} \frac{1}{2} (y_p - \hat{y}_p)^2, \quad (5)
$$

where $y_p$ and $\hat{y}_p$ are the $p$th calculated output and desired output, respectively.

If

$$
E > k_e \quad (6)
$$

To determine the proper number of rules or the structure of FNNs, we adopt a clustering algorithm in the portioning of input-output space. In order to obtain a compact structure, an incremental self-organized learning algorithm based on distance method is proposed here. The number of clusters is adjusted automatically during incremental learning, resulting in a flexible number of fuzzy partitioning of input-output space, as well as the optimal number of fuzzy rules. An initial structure of the FNN is constructed.

B. The rule generated algorithm is described below:

Step 0: Create the first rule by simply taking the position of the first example from the input data stream as the first rule centre $C_1$, and setting a value 0 for its width $\sigma_1$.

Step 1: If all examples of the data stream have been processed, the algorithm is finished. Else, the current example $x_i$ is taken and the distances between this example and all the $n$ already created cluster centres $C_j$ are calculated.

$$
D_j = \left\| x_i - c_j \right\|, \quad j = 1, 2, \ldots, n \quad (7)
$$

are calculated.

Step 2: If there is a rule center (centers) $C_j$ for $j = 1, 2, \ldots, n$, so that the distance value, $D_j = \left\| x_i - c_j \right\|$ is equal
to, or less than, the radius $\sigma_j$, it is assumed that the current example $x_i$ belongs to a rule $C_n$ with the minimum of these distances:

$$D_{im} = \|x_i - c_m\| = \min D_j,$$

where $D_j \leq \sigma_j, j = 1,2, ..., n$.

In this case, neither a new rule is created, nor any existing rule is updated (the cases of $x_4$ and $x_6$ in Figure 1) and the algorithm returns to Step 1.

Step 4: If $S_{ia}$ is not greater than $2\times k_d$ the cluster then $a$ is updated by moving its centre $C_a$, and increasing the value of its widths $\sigma_a$. The updated new widths $\sigma_a^{new}$ is set to be equal to $S_{ia}/2$ and the new centre $C_a^{new}$ is located on the line connecting the new input vector $x_i$ and the cluster centre $C_a$, so that the distance from the new centre $C_a^{new}$ to the point $x_i$ is equal to $\sigma_a^{new}$. The algorithm returns to Step 1.

Step 5: If $S_{ia}$ is greater than $2\times k_d$, the example $x_i$ does not belong to any existing rules. A new rule is created in the same way as described in Step 0, and the algorithm returns to Step 1.

C. Incremental learning of the ILSFNN

Suppose that fuzzy rules are generated according the criteria of rules generation. The network enters the incremental optimization-learning phase. Initialize parameters of the network with the values gained in the structure establishing phase. An improved back propagation (BP) algorithm [12,13] with adaptive learning rate and momentum is adopted for optimizing the parameters. Thereby the final network model is obtained.

Substituting Eq. (3) into Eq. (4) for an arbitrary input $x_p$ yields the output of network is

$$y_p = \sum_{k=1}^{l} \frac{\mu_k(x_p)w_k}{\sum_{k=1}^{l} \mu_k(x_p)}$$

According to Eqs.(5) yields

$$\frac{\partial E}{\partial c_a} = \frac{\partial E}{\partial y_p} \frac{\partial y_p}{\partial \mu_k} \frac{\partial \mu_k}{\partial c_a}$$

$$= \left( y_p - \hat{y}_p \right) \mu_k(x_p) \frac{2(x_p - c_a)}{\sigma_a^2}$$

According to Eqs.(13), then Eqs. (14) (15) (16) is obtained.

$$c_a(t+1) = \frac{n-1-l}{n} c_a(t) - \frac{1+l}{n} \sum_{m=1}^{n} \frac{\partial E}{\partial \mu_m} c_a(t) - \frac{1+l}{n} \sum_{m=1}^{n} \left( y_p - \hat{y}_p \right) w_m - y_p$$

$$\times \mu_k(x_p) \left( \frac{2(x_p - c_a)}{\sigma_a^2} \right)$$

$$\sigma_a(t+1) = \frac{n-1-l}{n} \sigma_a(t) - \frac{1+l}{n} \sum_{m=1}^{n} \frac{\partial E}{\partial \mu_m} \sigma_a(t) - \frac{1+l}{n} \sum_{m=1}^{n} \left( y_p - \hat{y}_p \right) w_m - y_p$$

$$\times \mu_k(x_p) \left( \frac{2(x_p - c_a)}{\sigma_a^2} \right)$$

$$w_k(t+1) = \frac{n-1-l}{n} w_k(t) - \frac{n+l}{n} \sum_{i=1}^{n} \frac{\partial E}{\partial w_k}$$

$$= \frac{n-1-l}{n} w_k(t) - \frac{n+l}{n} \sum_{i=1}^{n} \left( y_p - \hat{y}_p \right) \frac{\mu_i(x_p)}{\sum_{j=1}^{l} \mu_j(x_p)}$$

$$i = 1, \ldots, n \quad p = 1, \ldots, p \quad k = 1, \ldots, l,$$

Then $c_{ik}$, $\sigma_{ik}$ and $w_k$ are updated according to Eqs. (14)-(16), respectively.

Where $l$ is called the amnesic parameter which can further improve the convergence rate.

IV. SIMULATION STUDIES

In this section, the effectiveness of the proposed algorithm is demonstrated in the examples of function approximation. Here, the underlying function to be approximated is the Hermite polynomial:

$$f(x) = 1.1(1-x+2x^2) \exp(-\frac{x^2}{2})$$

Random sampling of the interval $[-4; 4]$ is used in obtaining 200 input–output data for the training set. The results are shown in Figure 2.
Through the incremental learning depicted in Section 3, seven fuzzy rules are obtained. The approximation error is measured by the root mean squared error (RMSE) calculated over another 200 uniformly sampled data in the same interval.

A comparison of structure and performance of different algorithms is shown in Table 1. From the table, one may note that there are fewer system parameters defined priori in our algorithm, which offer flexible trade-off between complexity and performance in other learning algorithms such as D-FNN.

oscillation problem and cause learning to be endless, so the robustness of the other algorithm is doubtful.

Table 1: Comparison of structure and performance of different algorithms

<table>
<thead>
<tr>
<th></th>
<th>ILSFN</th>
<th>D-FNN</th>
<th>OLS</th>
<th>M-RAN</th>
<th>RANEKFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of rules</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>Number of parameters</td>
<td>15</td>
<td>24</td>
<td>22</td>
<td>25</td>
<td>40</td>
</tr>
<tr>
<td>Epochs</td>
<td>200</td>
<td>200</td>
<td>7</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.0067</td>
<td>0.0054</td>
<td>0.009</td>
<td>-0.0438</td>
<td>-0.0262</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

In this paper, a new algorithm is proposed for creating a ILSFNN whose structure is automatically adapted via declines dramatically by comparison. A hierarchical on-line self-organizing learning scheme is employed to configure fuzzy systems from sample data so that parameters and structure can be done automatically and simultaneously without partitioning the input space. There are very few system parameters to be defined priori, so the complexity is reduced, and the structure identification can be speed up.

VI. ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for their high quality reviews and valuable comments. This project is supported by Scientific Research Fund of Hunan Provincial Education Department (11C0009).

VII. REFERENCES