Genetic Design of Fuzzy Neural Networks Based on Respective Input Spaces Using Interval Type-2 Fuzzy Set

Keon-Jun Park¹ and Dong-Yoon Lee²*

¹Dept. of Information and Communication Engineering, Wonkwang University, 344-2, Shinyong-dong, Iksan-si, Chonbuk, 570-749 South Korea
²Dept. of Electrical Electronic Engineering, Joongbu University, 101, Daehak-ro, Chubu-myeon, Geumsan-gun, Chungnam, 312-702 South Korea
bird75@wonkwang.ac.kr, dylee@joongbu.ac.kr

Abstract

In this paper, we propose the genetic design of fuzzy neural networks with multi-output based on interval type-2 fuzzy set (IT2FSFNNm) for pattern recognition. IT2FSFNNm is the networks of combination between the fuzzy neural networks (FNNs) and interval type-2 fuzzy set with uncertainty. The premise part of the networks is composed of the fuzzy partition of respective input spaces and the consequence part of the networks is represented by polynomial functions with interval set. We also consider real-coded genetic algorithms to estimate the optimal values of the parameters of IT2FSFNNm. The numerical experimentation is used for evaluating the proposed networks for pattern recognition.

Keywords: Fuzzy Neural Networks (FNNs), Interval Type-2 Fuzzy Set (IT2FS), Multiple-Output, Pattern Recognition, Genetic Algorithms (GAs)

1. Introduction

Fuzzy neural networks (FNNs) [1, 2] are hybrid models that combine the human-like reasoning method of fuzzy inference systems and connectionist structure of neural networks. There are still many approaches to apply to FNNs [3-5]. Typically, FNNs are represented by fuzzy “if-then” rules while the back propagation (BP) is used to optimize the parameters of the networks.

The concept of a type-2 fuzzy set as an extension of fuzzy sets of type-1 is introduced by Zadeh [6]. Mizumoto and Tanaka [7] studied the set theoretic operations of type-2 fuzzy sets and discussed properties of membership grades of such sets. In contrast to type-1 fuzzy sets, in type-2 fuzzy sets the membership grades are not numeric values but fuzzy sets defined in the unit interval. Mendel and Karnik [8-10] studied the theory of type-2 fuzzy logic systems. These are also described in the form of fuzzy if-then rules, but their premises and/or consequents are type-2 fuzzy sets. An interval type-2 fuzzy neural network (IT2FNN) comes as a result of symbiotic interaction of interval type-2 fuzzy set and neural networks.

In this paper, we propose the design of fuzzy neural networks with multi-output based on interval type-2 fuzzy set (IT2FSFNNm) that consists of interval type-2 fuzzy set forming the premise part of the rules and neural networks viewed as the consequence part. The premise part of the networks is composed of the fuzzy grid partition by using respective input space. We use interval-valued triangular membership functions while the coefficients of the polynomial functions with interval set located in the consequents of the rules are learned by

* Corresponding author: dylee@joongbu.ac.kr
BP algorithm. We also optimize the parameters of the networks such as the apexes of membership functions, uncertainty coefficient, the learning rate, and the momentum coefficient using real-coded genetic algorithms [11]. The proposed networks are evaluated through the numeric experimentation for pattern recognition.

The paper is organized as follows. In Section 2, the design of the SOFNN is described. The optimization of the proposed networks is organized in Section 3. Simulations of IT2FSFNNm are compared with other algorithms in Section 4. Finally, conclusions are summarized in Section 5.

2. Design of IT2FSFNNm

The structure of IT2FSFNNm emerges at the junction of interval type-2 fuzzy sets and fuzzy neural networks. In this section, the form of fuzzy if-then rules along with their development mechanism is discussed.

2.1. Interval Type-2 Fuzzy Set

A type-2 fuzzy set, denoted here by \(\widetilde{A}\), is characterized by a type-2 membership function \(\mu_\widetilde{A}(x)\) of the form

\[
\widetilde{A} = \int_{x \in X} \mu_\widetilde{A}(x) / x = \int_{x \in X} \left[ \int_{u \in J_x} f_x(u) / u \right] / x, J \subseteq [0,1].
\]  

(1)

The domain of a secondary membership function is called the primary membership of \(x\). In (1), \(J_x\) is the primary membership of \(x\), where \(J_x \subseteq [0,1]\) for \(\forall x \in X\). The amplitude of a secondary membership function is called a secondary grade. In (1), \(f_x(u)\) stands for a secondary grade.

When \(f_x(u) = 1, \forall u \in J_x \subseteq [0,1]\), the secondary membership functions are intervals and \(\mu_\widetilde{A}(x)\) is referred to as an interval type-2 membership function. Therefore, the type-2 fuzzy set \(\widetilde{A}\) can be re-expressed as

\[
\widetilde{A} = \int_{x \in X} \mu_\widetilde{A}(x) / x = \int_{x \in X} \left[ 1 / u \right] / x, J_x \subseteq [0,1].
\]  

(2)

Uncertainty present in the primary membership values of the type-2 fuzzy set, consist of a bounded region - footprint of uncertainty (FOU). The FOU is shaded uniformly to underline the interval nature of this type-2 fuzzy set. Here uncertainty resides with the apexes of the linear segments of the membership functions whose values could vary within a certain range and its spreads are adjusted by \((1 + \rho)\sigma\) using the uncertainty parameter \(\rho\).

An upper membership function and a lower membership function are two type-1 membership functions that form the bounds for the FOU of the type-2 fuzzy set. The upper membership function is denoted here by \(\mu_\widetilde{A}(x)\). The lower membership function is described by \(\mu_\widetilde{A}(x)\). Hence, (2) can be rewritten in the following form:
\[
\tilde{A} = \int_{x \in X} \left[ \int_{u \in \bar{\mu}_i(x)} \frac{1}{u} \right] / X.
\] (3)

2.2. The Structure of IT2FSFNNm

As mentioned earlier, the structure of the IT2FSFNNm involves interval type-2 fuzzy sets in the premise part and neural networks present in the consequence part of the rules. The overall topology of the network is illustrated in Figure 1.

![Figure 1. The Structure of IT2FSFNNm](image)

The notation used in this figure requires some clarification. The circles denote nodes of the IT2FSFNNm while the node ‘N’ pertains to a normalization procedure applied to the membership grades of the input variable. The TR denote nodes of the type-reduction and the output \( \hat{y} \) of the ‘\( \sum \)’ is governed by some nonlinear function.

IT2FSFNNm is implied by the fuzzy partition of respective input spaces. In this sense, each rule can be viewed as a certain rule of the following format.

\[
R^{kc} : \text{If } x_k \text{ is } \tilde{A}^{kc} \text{ Then } y^{kc} = f(x_1, \ldots, x_d)
\] (4)

As far as inference schemes are concerned, we distinguish between three cases.

Case 1 (Simplified Inference):

\[
f = W_{kc}^{t0}
\] (5)

Case 2 (Linear Inference):

\[
f = W_{kc}^{s0} + \sum_{i=1}^{d} W_{kc}^{si} \cdot x_i
\] (6)

Case 3 (Modified Quadratic Inference):

\[
f = W_{kc}^{t0} + \sum_{i=1}^{d} W_{kc}^{si} \cdot x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} W_{kc}^{z} \cdot x_i x_j,
\]

where, \( z = d + 1, \ldots, d(d + 1)/2 \)

(7)
To be more specific, $R^k_c$ is the $k$, $c$-th fuzzy rule, while $A^k_{c_{k_c}}$ denotes $k$, $c$-th interval type-2 fuzzy set. $W^u_{k_{k_c}} = [w^u_{k_{k_c}} - s^u_{k_{k_c}}, w^u_{k_{k_c}} + s^u_{k_{k_c}}]$, $i = 0, \ldots, d(d+1)/2$ are consequent parameters of the rules. $w^u_{k_{k_c}}$ and $s^u_{k_{k_c}}$ are the center and the spread of $W^u_{k_{k_c}}$, respectively.

The functionality of each layer is described as follows.

[Layer 1] The nodes in this layer transfer the inputs to the respective inputs.
[Layer 2] The nodes here are used to calculate the membership degrees for given membership functions. When interval type-2 fuzzy sets are used, the firing set $f^k_c$ of the rule becomes an interval coming in the following form.

$$f^k_c = [\mu^k_c, \mu^k_c]$$  \hspace{1cm} (8)

[Layer 3] The nodes in this layer normalize the membership degrees for each input.

$$\hat{f}^k_c = [\hat{f}^k_c, \hat{f}^k_c]$$

$$\bar{f}^k_c = \frac{\mu^k_c}{\sum_{g=1}^{G} \mu^k_c}, \hat{f}^k_c = \frac{\mu^k_c}{\sum_{g=1}^{G} \mu^k_c}$$  \hspace{1cm} (9)

[Layer 4] The nodes in this layer are used to conduct type-reduction. Note that left-most point $y_{sl}$ and right-most point $y_{sr}$ depend upon the values of $\hat{f}^k_c$. Hence, $y_{sl}$ and $y_{sr}$, using Karnik-Mendel (KM) algorithm can be expressed as the follows.

$$y_{sl} = \frac{\sum_{k=1}^{d} \sum_{g=1}^{G} \hat{f}^k_c \hat{f}^k_c \hat{y}^k_{sl}}{\sum_{k=1}^{d} \sum_{g=1}^{G} \hat{f}^k_c \hat{f}^k_c}, \ y_{sr} = \frac{\sum_{k=1}^{d} \sum_{g=1}^{G} \hat{f}^k_c \hat{f}^k_c \hat{y}^k_{sr}}{\sum_{k=1}^{d} \sum_{g=1}^{G} \hat{f}^k_c \hat{f}^k_c}$$  \hspace{1cm} (10)

Here, $\hat{f}^k_c$ and $\hat{f}^k_c$ are upper and lower firing sets which have an effect on $y_{sl}$ and $y_{sr}$, respectively.

[Layer 5] The nodes in this layer compute the overall outputs.

In the IT2FSFNNm, the output $\hat{y}_s$ is an interval set, so we require to defuzzify (decode) it by taking the average of $y_{sl}$ and $y_{sr}$. Commonly, the defuzzified output of IT2FSFNNm is computed as

$$\hat{y}_s = \frac{y_{sl} + y_{sr}}{2}.$$  \hspace{1cm} (12)

2.3. Learning Algorithm

The parametric learning of the IT2FSFNNm is realized by adjusting connections of the neurons and as such it could be realized by running a standard Back-Propagation (BP) algorithm. The performance index $E_p$ is based on the Euclidean distance,

$$E_p = \frac{1}{2} \sum_{s=1}^{q} (y_{ps} - \hat{y}_{ps})^2.$$  \hspace{1cm} (13)
where, $E_p$ is an error reported for the $p$-th data, $y_{ps}$ is the $s$, $p$-th target output data and $\hat{y}_{ps}$ stands for the $s$, $p$-th actual output of the model.

As far as learning is concerned, the connections are changed (adjusted) in a standard fashion,

$$w_{kc}^{s0}(p+1) = w_{kc}^{s0}(p) + \Delta w_{kc}^{s0}$$

(14)

where this update formula follows the gradient descent method, namely

$$\Delta w_{kc}^{s0} = \eta \left( -\frac{\partial E_p}{\partial w_{kc}^{s0}} \right)$$

(15)

with $\eta$ being a positive learning rate.

From the chain rules we have the following expression.

$$-\frac{\partial E_p}{\partial w_{kc}^{s0}} = -\frac{1}{2} \frac{\partial E_p}{\partial \hat{y}_{ps}} \left[ \frac{\partial \hat{y}_{ps}}{\partial \hat{y}_{sl}} + \frac{\partial \hat{y}_{ps}}{\partial \hat{y}_{sr}} + \frac{\partial \hat{y}_{ps}}{\partial w_{kc}^{s0}} \frac{\partial w_{kc}^{s0}}{\partial w_{kc}^{s0}} + \frac{\partial \hat{y}_{ps}}{\partial w_{kc}^{s0}} \frac{\partial w_{kc}^{s0}}{\partial w_{kc}^{s0}} \right]$$

(16)

Quite commonly to accelerate convergence, a momentum coefficient $\alpha$ is being added to the learning expression. And $s_{kc}^{s0}$ is obtained in the same way as $w_{kc}^{s0}$. Then the complete update formula reads as follows.

$$\Delta w_{kc}^{s0} = 0.25\eta \left( y_{ps} - \hat{y}_{ps} \right) \left( \hat{f}_{kc}^a + \hat{f}_{kc}^w \right) + \alpha \left( w_{kc}^{s0}(p) - w_{kc}^{s0}(p-1) \right)$$

(17)

$$\Delta s_{kc}^{s0} = 0.25\eta \left( y_{ps} - \hat{y}_{ps} \right) \left( \hat{f}_{kc}^a + \hat{f}_{kc}^w \right) s_i + \alpha \left( s_{kc}^{s0}(p) - s_{kc}^{s0}(p-1) \right)$$

(18)

$$\Delta w_{kc}^{s1} = 0.25\eta \left( y_{ps} - \hat{y}_{ps} \right) \left( \hat{f}_{kc}^a + \hat{f}_{kc}^w \right) k_i + \alpha \left( w_{kc}^{s1}(p) - w_{kc}^{s1}(p-1) \right)$$

(19)

$$\Delta s_{kc}^{s1} = 0.25\eta \left( y_{ps} - \hat{y}_{ps} \right) \left( \hat{f}_{kc}^a + \hat{f}_{kc}^w \right) s_j + \alpha \left( s_{kc}^{s1}(p) - s_{kc}^{s1}(p-1) \right)$$

(20)

$$\Delta w_{kc}^{s2} = 0.25\eta \left( y_{ps} - \hat{y}_{ps} \right) \left( \hat{f}_{kc}^a + \hat{f}_{kc}^w \right) x_i \eta + \alpha \left( w_{kc}^{s2}(p) - w_{kc}^{s2}(p-1) \right)$$

(21)

$$\Delta s_{kc}^{s2} = 0.25\eta \left( y_{ps} - \hat{y}_{ps} \right) \left( \hat{f}_{kc}^a + \hat{f}_{kc}^w \right) x_j \eta + \alpha \left( s_{kc}^{s2}(p) - s_{kc}^{s2}(p-1) \right)$$

(22)

3. Genetic Optimization of IT2FSFNNm

It has been demonstrated that genetic algorithms (GAs) [11] are useful global population-based optimizers. GAs are shown to support robust search in complex search spaces. Given their stochastic character, such methods are less likely to get trapped in local minima (which becomes quite a common problem in case of gradient-descent techniques). The search in the solution space is completed with the aid of several genetic operators with reproduction, crossover, and mutation being the standard ones.

Let us briefly recall the essence of these operators. Reproduction is a process in which the mating pool for the next generation is chosen. Individual strings are copied into the mating pool according to the values of their fitness functions. Crossover usually proceeds in two steps. First, members from the mating pool are mated at random. Secondly, each pair of strings undergoes crossover as follows; a position $l$ along the string is selected uniformly at random from the interval $[1, l-1]$, where $l$ is the length of the string. Swapping all characters between the positions $k$ and $l$ creates two new strings. Mutation is a random alteration of the value of a string position. In real coding, mutation is defined as an alternation at a random value in special boundary. Usually mutation occurs with a small probability. Those operators,
combined with the proper definition of the fitness function, constitute the main body of the genetic optimization. A general flowchart of the genetic algorithm is shown in Figure 2.

![Figure 2. A General GA Flowchart](image)

In this study, in order to optimize the parameters of the IT2FSFNNm, we determined the parameters of the membership function and uncertainty parameter of the premise part and the learning rate and the momentum coefficient occurring in the conclusion part. Figure 3 visualizes an arrangement of the content of the chromosome to be used in genetic optimization.

![Figure 3. Data Structure of Chromosomes](image)

### 4. Experimental Studies

In this section, we use the Iris dataset [12]. The Iris dataset is a collection of 150 Iris flowers of 3 kinds, with four attributes, leaf and petal width and length in cm. Three classes are the setosa, versicolor, and virginica.

For the evaluation of the performance of the network, the random sub-sampling method was applied. The random sub-sampling was performed with 5 data splits of the data set ($K=5$). Each split was randomly selected from the training examples and the test examples with the ratio of 7:3.

The classification ratio ($CR$) is defined as the average of the separate estimates $E_p$.

\[
E_p = \frac{\text{No. of classification}}{\text{No. of test examples}} \times 100 \tag{23}
\]

\[
CR = \frac{1}{K} \sum_{i=1}^{K} E_p \tag{24}
\]

Another performance index ($PI$) is based on the Mean Squared Error (MSE).

\[
PI = \frac{1}{K} \sum_{i=1}^{K} MSE \tag{25}
\]
We experimented with the proposed network using the parameters outlined in Table 1.

**Table 1. Initial Parameters for GAs**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation</td>
<td>100</td>
</tr>
<tr>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.65</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.1</td>
</tr>
<tr>
<td>Uncertainty coefficient</td>
<td>$-1.0 \leq \rho \leq 1.0$</td>
</tr>
<tr>
<td>Learning rate</td>
<td>$0.0 \leq \eta \leq 0.01$</td>
</tr>
<tr>
<td>Moment coefficient</td>
<td>$0.0 \leq \alpha \leq 0.001$</td>
</tr>
</tbody>
</table>

Table 2 summarizes the performance for IT2FSFNNm before optimization and Table 3 shows the performance for IT2FSFNNm using genetic optimization. From these tables we know that the optimized IT2FSFNNm is better than before optimization. From Table 3 we select the network that has twenty fuzzy rules and linear inference engine. This network exhibits $CR=99.05\pm0.67$, $PI=0.020\pm0.00$ for training datasets and $CR=99.56\pm0.99$, $PI=0.014\pm0.00$ for testing datasets.

Table 4 shows the confusion matrix for the training and testing data set. From Table 4, the results show some misclassification. In the training data, versicolor is classified as virginica with $2.29\pm1.28$ error ratio and virginica is classified as versicolor with $0.57\pm1.28$ error ratio. In the testing data, versicolor is classified as virginica with $1.33\pm2.98$ error ratio.

**Table 2. Performance of the IT2FSFNNm**

<table>
<thead>
<tr>
<th>No. of MFs</th>
<th>Inference (Case)</th>
<th>CR Training</th>
<th>CR Testing</th>
<th>PI Training</th>
<th>PI Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>85.14±0.98</td>
<td>83.56±3.72</td>
<td>0.092±0.01</td>
<td>0.089±0.01</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>97.14±0.95</td>
<td>96.00±2.43</td>
<td>0.038±0.00</td>
<td>0.037±0.01</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>96.00±0.43</td>
<td>97.78±1.57</td>
<td>0.036±0.00</td>
<td>0.038±0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>97.90±0.80</td>
<td>96.00±2.43</td>
<td>0.026±0.00</td>
<td>0.027±0.00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>97.71±0.52</td>
<td>96.44±1.99</td>
<td>0.024±0.00</td>
<td>0.027±0.01</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>98.29±1.04</td>
<td>97.78±1.57</td>
<td>0.024±0.00</td>
<td>0.029±0.00</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>96.38±1.24</td>
<td>96.89±1.99</td>
<td>0.020±0.00</td>
<td>0.019±0.01</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>98.48±0.52</td>
<td>95.56±2.22</td>
<td>0.016±0.00</td>
<td>0.023±0.01</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>97.71±0.52</td>
<td>95.11±2.90</td>
<td>0.015±0.00</td>
<td>0.025±0.01</td>
</tr>
</tbody>
</table>

**Table 3. Performance of the Optimized IT2FSFNNm**

<table>
<thead>
<tr>
<th>No. of MFs</th>
<th>Inference (Case)</th>
<th>CR Training</th>
<th>CR Testing</th>
<th>PI Training</th>
<th>PI Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>96.38±0.80</td>
<td>97.33±1.86</td>
<td>0.032±0.01</td>
<td>0.030±0.01</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>99.24±0.80</td>
<td>96.89±1.22</td>
<td>0.025±0.00</td>
<td>0.033±0.01</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>99.24±0.43</td>
<td>99.11±1.22</td>
<td>0.028±0.00</td>
<td>0.028±0.00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>98.10±0.67</td>
<td>99.11±1.22</td>
<td>0.023±0.00</td>
<td>0.017±0.01</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>99.81±0.43</td>
<td>98.22±0.99</td>
<td>0.021±0.00</td>
<td>0.019±0.00</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>99.62±0.52</td>
<td>98.67±1.22</td>
<td>0.021±0.00</td>
<td>0.019±0.00</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>98.29±0.43</td>
<td>99.56±0.99</td>
<td>0.022±0.00</td>
<td>0.020±0.00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>99.05±0.67</td>
<td>99.56±0.99</td>
<td>0.020±0.00</td>
<td>0.014±0.00</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>99.62±0.52</td>
<td>98.67±1.22</td>
<td>0.018±0.00</td>
<td>0.019±0.01</td>
</tr>
</tbody>
</table>
Table 4. Performance Evaluation by Confusion Matrix

(a) Training data

<table>
<thead>
<tr>
<th></th>
<th>Setosa</th>
<th>Versicolor</th>
<th>Virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setosa</td>
<td>100.00±0.00</td>
<td>0.00±0.00</td>
<td>0.00±0.00</td>
</tr>
<tr>
<td>Versicolor</td>
<td>0.00±0.00</td>
<td>97.71±1.28</td>
<td>2.29±1.28</td>
</tr>
<tr>
<td>Virginica</td>
<td>0.00±0.00</td>
<td>0.57±1.28</td>
<td>99.43±1.28</td>
</tr>
</tbody>
</table>

(b) Testing data

<table>
<thead>
<tr>
<th></th>
<th>Setosa</th>
<th>Versicolor</th>
<th>Virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setosa</td>
<td>100.00±0.00</td>
<td>0.00±0.00</td>
<td>0.00±0.00</td>
</tr>
<tr>
<td>Versicolor</td>
<td>0.00±0.00</td>
<td>98.67±2.98</td>
<td>1.33±2.98</td>
</tr>
<tr>
<td>Virginica</td>
<td>0.00±0.00</td>
<td>0.00±0.00</td>
<td>100.00±0.00</td>
</tr>
</tbody>
</table>

Figure 4 presents the optimization procedure for the CR and PI when using twenty rules with Case 2 (Linear Inference) obtained in successive generations of the genetic optimization. These figures depict the average values using the random sub-sampling.

Figure 4. Optimization Process for the Selected Network

The performance of the proposed network is compared with the performance of some other models reported in the literature; refer to Table 5. The comparison shows that the proposed model outperforms several previous developed models.

Table 5. Comparison of Performance with Previous Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Classification Ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEFCLASS [13]</td>
<td>96.0</td>
</tr>
<tr>
<td>C4.5 [14]</td>
<td>94.0</td>
</tr>
<tr>
<td>FID3.1 [15]</td>
<td>96.0</td>
</tr>
<tr>
<td>HNFB [16]</td>
<td>98.67</td>
</tr>
<tr>
<td>HNFQ [17]</td>
<td>98.67</td>
</tr>
<tr>
<td>HNFB-1 [18]</td>
<td>98.67</td>
</tr>
<tr>
<td>Our model</td>
<td>99.56</td>
</tr>
</tbody>
</table>
5. Conclusion

In this paper, we have presented the design of fuzzy neural networks based on interval type-2 fuzzy set with multiple-output for pattern recognition and discussed its optimization using real-coded genetic algorithms.

The input spaces of the proposed networks were divided as the form of the fuzzy grid partition by using respective input space to generate the fuzzy rules. The parameters of the networks such as the apexes of membership functions, uncertainty coefficient, the learning rate, and the momentum coefficient are optimized using real-coded genetic algorithms.

From the result in the previous section, we were able to design good networks for pattern recognition. Through the use of performance we were able to achieve a balance between the approximation and generalization abilities of the resulting network. Finally it could be possible to apply to many fields.

References

Authors

Keon-Jun Park, received the B.S degree in Electrical & Electronics Engineering from the Wonkwang University, Iksan, South Korea, in 2003, and the M.S degree in Control and Instrumentation Engineering in 2005. He received the Ph.D degree in Electrical Engineering from University of Suwon, Suwon, South Korea, in 2010. He is currently a research professor with Information & Communication Engineering, Wonkwang University, Korea. His research interests include computational intelligence, intelligence modeling and control, and pattern recognition.

Dong-Yoon Lee, received the M.S degree in Electrical Engineering and the Ph.D degree in Electrical Electronic Engineering from Yonsei University, Seoul, South Korea, in 1990 and 2001, respectively. He is currently a professor with Joongbu University, Chungnam, South Korea. His research interests include security system, artificial intelligence and application.