International Journal of Grid and Distributed Computing
Vol. 5, No. 3, September, 2012

Optimal Design of Fuzzy Clustering-based Fuzzy Neural Networks for Pattern Classification

Keon-Jun Park[1], Jong-Pil Lee[2] and Dong-Yoon Lee[3]

[1]*Dept. of Information and Communication Engineering, Wonkwang University, 344-2, Shinyong-dong, Iksan-si, Chonbuk, 570-749 South Korea*

[2]*Korea Electric Institute, 147-11 Sachang-dong, Heungdeok-gu, Cheongju-city, Chungbuk, 361-831 South Korea*

[3]*Dept. of Electrical Electronic Engineering, Joongbu University, 101, Daehak-ro, Chubu -myeon, Geumsan-gun, Chungnam, 312-702 South Korea*

birds@suwon.ac.kr, jp1212@paran.com, dylee@joongbu.ac.kr

Abstract

We introduce a new category of fuzzy neural networks with multiple-output based on fuzzy clustering algorithm, especially, fuzzy c-means clustering algorithm (FCM-based FNNm) for pattern classification in this paper. The premise part of the rules of the proposed networks is realized with the aid of the scatter partition of input space generated by FCM clustering algorithm. The partitioned local spaces describe the fuzzy rules and the number of the partitioned local spaces is equal to the number of clusters. Due to these characteristics, we may alleviate the problem of the curse of dimensionality. The consequence part of the rules is represented by polynomial functions with multiple-output for pattern classification. And the coefficients of the polynomial functions are learned by back propagation algorithm. To optimize the parameters of the proposed FCM-based FNNm we consider real-coded genetic algorithms. The proposed networks are evaluated with the use of numerical experimentation.

Keywords: *Fuzzy Neural Networks, FCM clustering algorithm, Scatter partition of input space, Optimization, Genetic Algorithms*

1. Introduction

Fuzzy neural networks (FNNs) [1, 2] have emerged as one of the active areas of research in fuzzy inference systems and neural networks. FNNs are predominantly concerned with the integration of these two fields. Fuzzy inference systems have been studied to model uncertain and/or ambiguous characteristics inherent to experimental data. Fuzzy inference systems are good at explaining decisions but they cannot directly obtain the fuzzy rules. Neural networks are good at classifying patterns but they are not good at explaining how they acquire decisions. These limitations have been the major points and these techniques are combined to overcome the shortcomings. Since its inception, the research of FNNs has been a focal point of various endeavors and has demonstrated many fruitful results in application [3, 4, 5]. Typically, FNNs are represented by fuzzy “if-then” rules while the back propagation (BP) is used to optimize the parameters.

The generation of the fuzzy rules and the adjustment of its membership functions were done by trial and error and/or operator’s experience. The designers find it difficult to develop adequate fuzzy rules and membership functions to reflect the essence of the data. Moreover, some information gets lost or ignored on purpose when human operators articulate their experience in the form of linguistic rules. As a consequence, there is a need for an optimization environment to construct and/or adjust a collection of linguistic rules.

In this paper, we present the structure of fuzzy neural networks with multiple-output by means of fuzzy c-means clustering algorithm [6] (FCM-based FNNm). The premise part of the rules is realized with the aid of the scatter partition of input space generated by FCM clustering algorithm. The partitioned local spaces describe the fuzzy rules. The consequence part of the rules is represented by polynomial functions with multiple-output for pattern classification. And the coefficients of the polynomial functions are learned by BP algorithm. We also optimize the parameters of the networks using real-coded genetic algorithms [7]. The proposed networks are evaluated through the numeric experimentation.

The paper is organized as follows. Section 2 is concerned with the design of FCM-based FNNm. Section 3 deals with the optimization of FCM-based FNNm. Section 4 presents results of numeric experimentations. Finally Section 5 concludes the paper. $\sum u_{ip} = 1,\ 1 \le p \le N$

2. Design of Fuzzy c-means Clustering-based Fuzzy Neural Networks

In this section, the form of fuzzy clustering if-then rules along with their development mechanism is discussed. More specifically, we elaborate on the three types of fuzzy inference and present the learning algorithm.

2.1. Fuzzy c-means Clustering Algorithm

The premise part of the FCM-based FNNm is developed by means of the Fuzzy C-Means clustering algorithm [6] as the fuzzy clustering. This algorithm is aimed at the formation of 'c' fuzzy sets (relations) in $\mathbf{R}^n$. Consider the set $\mathbf{X}$, which consists of N data points treated as vectors located in some n-dimensional Euclidean space. In clustering we assign patterns $\mathbf{x}_p \in \mathbf{X}$ into c clusters, which are represented by its prototypes $\mathbf{v}_i \in \mathbf{R}^n$. The assignment to individual clusters is expressed in terms of the partition matrix $\mathbf{U} = [u_{ip}]$ where

$$0 < \sum_{p=1}^{N} u_{ip} < N,\quad 1 \le i \le c \tag{1}$$

$$\sum_{i=1}^{c} u_{ip} = 1,\quad 1 \le p \le N \tag{2}$$

The objective function Q guiding the clustering is expressed as a sum of the distances of individual data from the prototypes $\mathbf{v}_1$, $\mathbf{v}_2$, ..., and $\mathbf{v}_c$,

$$Q = \sum_{i=1}^{c}\sum_{p=1}^{N} u_{ip}^{m} \left\| \mathbf{x}_p - \mathbf{v}_i \right\|^2 . \tag{3}$$

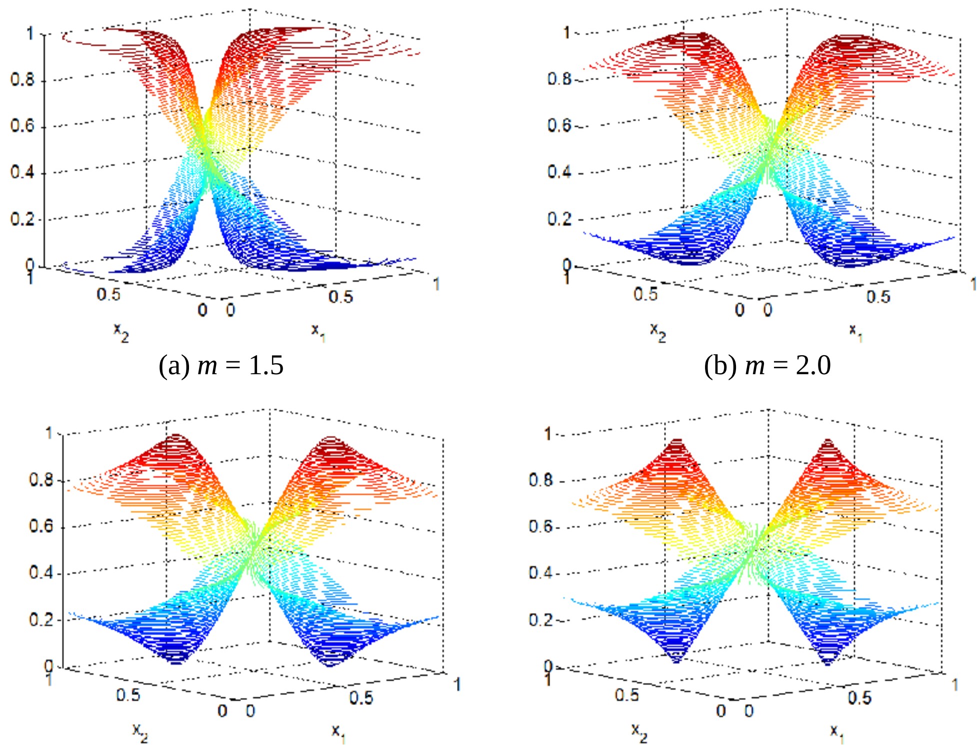
Here $\|\cdot\|$ denotes the Euclidean distance; 'm' stands for a fuzzification coefficient, m_i>1.0. The resulting partition matrix is denoted by $\mathbf{U} = [u_{ip}]$.

The minimization of Q is realized through successive iterations by adjusting both the prototypes and entries of the partition matrix, that is min $Q(\mathbf{U}, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_c)$. The corresponding formulas used in an iterative fashion read as follows.

$$\mathbf{v}_i = \frac{\sum_{p=1}^{N} u_{ip}^{m} \mathbf{x}_p}{\sum_{p=1}^{N} u_{ip}^{m}} \qquad (4)$$

$$u_{ip} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|\mathbf{x}_p - \mathbf{v}_i\|}{\|\mathbf{x}_p - \mathbf{v}_j\|} \right)^{\frac{2}{m-1}}} . \qquad (5)$$

Figure 1 shows the membership matrix according to fuzzification coefficient. The shape of membership grade is affected by the values of the fuzzification coefficient. This means that the shape of membership grade becomes sharper as the value of coefficient is larger.



(a) m = 1.5 (b) m = 2.0

(c) m = 2.5 (d) m = 3.0

Figure 1. Membership Matrix According to Fuzzification Coefficient

2.2. Structure of FCM-based FNNm

The structure of the FCM-based FNNm involves FCM clustering algorithm in the premise part and neural networks present in the consequence part of the rules. The overall topology of the networks is illustrated in Figure 2.

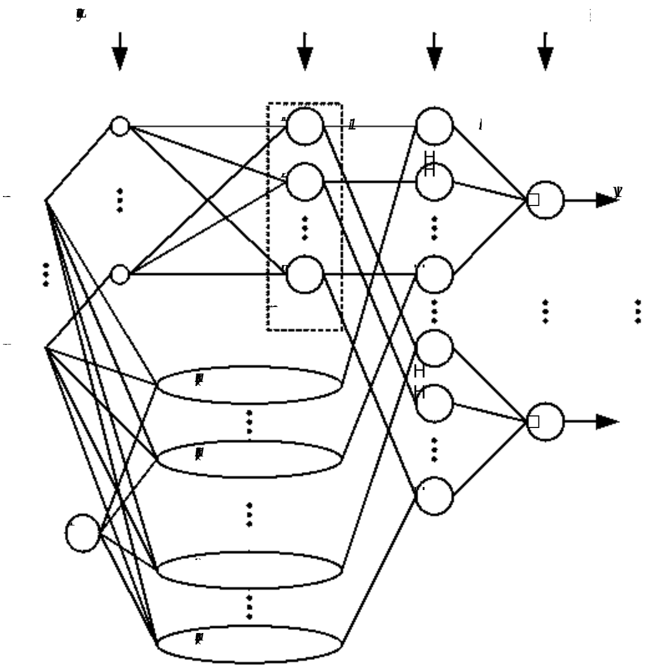


Figure 2. The Structure of FCM-based FNNm

$y\ q_{j\ k\ k}$ FCM-based FNNm is implied by the fuzzy scatter partition of input space. In this sense, each rule can be viewed as a certain rule of the following format

. (6)

As far as inference schemes are concerned, we distinguish these types Type 1 (Simplified Inference):

$d \quad d$

$s \quad s$

$fv \quad w$

$s \quad + \square \square\ _{wxx},\ z = {}_{d} + {}_{1,} \square\ d\,({}_{d} + {}_{1)} / {}_{2.}$

$p \quad k \quad k$

Type 2 (Linear Inference):

Type 3 (Modified Quadratic Inference):

(9)

To be more specific, is the *j*-th fuzzy rule, while denotes *j*-th membership grades using FCM clustering algorithm. , are consequent parameters of the rule and *s* is the number of output.

(7)

(8)

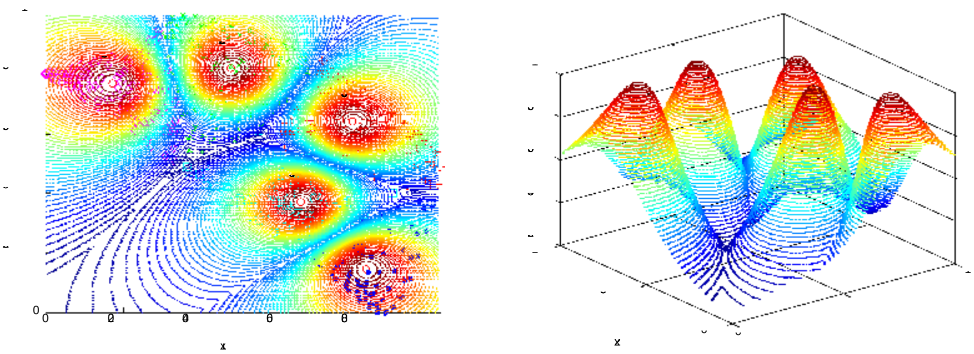
The functionality of each layer is described as follows.

[*Layer 1*] The nodes in this layer transfer the inputs.

[*Layer 2*] The nodes here are used to calculate the membership degrees using FCM clustering algorithm. From this algorithm, the firing strengths are as follows.

(10)

Figure 3 visualizes the example of the fuzzy partitioned spaces of input space with five clusters by means of FCM clustering algorithm.



(a) Partitioned local spaces (b) Membership grade

Figure 3. Scatter Partition of Input Space

h_j

[*Layer 3*] The nodes in this layer realize a certain inference process. (11)

$$h_{sj} \quad \sum_{j=1}^{n} \quad y \quad ys \quad 1$$

(12)

Where, all the entries of sum up to 1 as indicated by (1).

[*Layer 4*] The nodes in this layer compute the outputs.

n

$1 \quad E_y \quad 2$

2.3. Learning Algorithm

The parametric learning of the network 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 is based on the Euclidean distance,

(13)

ˆ

y_{ps}

where, E_P is an error reported for the p-th data, y_{ps} is the p ,s-th target output data and stands for the p, s-th actual output of the network.

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

(14)

where this update formula follows the gradient descent method.

(15)

with being a positive learning rate.

If the type of consequence part is simplified inference then . From the chain rules we have the following expression.

h_{sj}

(16)

(17)

(18)

(19)

Quite commonly to accelerate convergence, a momentum coefficient is being added to the learning expression. Then the complete update formula reads as follows

(20)

(21)

(22)

3. Optimization of Networks

The need to solve optimization problems arises in many fields and is especially dominant in the engineering environment. There are several analytic and numerical optimization techniques, but there are still large classes of functions that are fully addressed by these techniques. Especially, the standard gradient-based optimization techniques that are being used mostly at the present time are augmented by a differential method of solving search problems for optimization processes. Therefore, the optimization of fuzzy models may not be fully supported by the standard gradient-based optimization techniques, because of the nonlinearity of models represented by rules based on linguistic levels. This forces us to explore other optimization techniques such as genetic algorithms (GAs) [7].

It has been demonstrated that genetic algorithms are useful in a global population-based optimization. GAs are shown to support robust search in complex search spaces. Given their stochastic character, these methods are less likely to get trapped in local minima in comparison to the performance offered by gradient-descent techniques.

GAs start with a randomly generated population of *l* chromosomes positioned in solution (parameter) space. The population is evolved repeatedly toward achieving a better overall fitness value. The search in the solution space is completed with the aid of several genetic operators. There are three main generic genetic operators such as reproduction, crossover, and mutation supporting movements in the search space. Let us briefly elaborate on the essence of these operators.

Reproduction is a process in which the mating pool for the next generation becomes selected. Individual strings are copied into the mating pool according to their fitness function values.

Crossover usually proceeds in two steps. Firstly, 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. Mutation occurs with a small probability. Those operators, combined with the proper definition of the fitness function, constitute the main body of the genetic computing. A general flowchart of the genetic algorithm is shown in Figure 4.

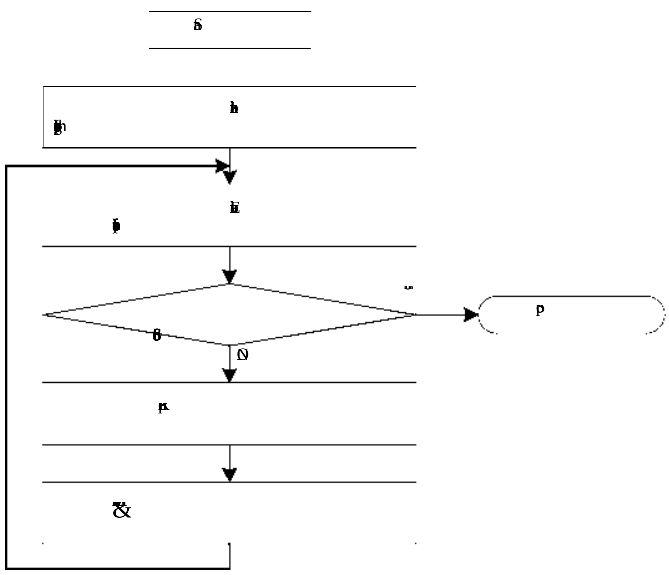


Figure 4. A General GA Flowchart

In order to optimize the parameters of the FCM-based FNNm, we determined the fuzzification coefficients associated with the corresponding clusters, the learning rate, and the momentum coefficient as the parameters. These parameters are genetically optimized across generation. Figure 5 illustrates an arrangement of the content of the chromosomes. Each chromosome is coded using real numbers (instead of binary numbers). This type of coding is helpful from the point of view of effectiveness of the overall search process.

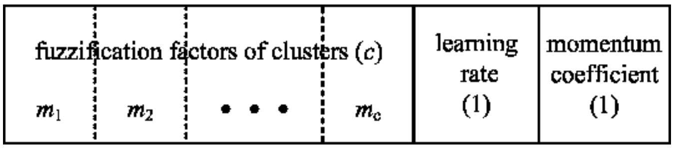


Figure 5. Data Structure of Chromosomes

4. Experimental Studies

We discuss three numerical examples in order to evaluate the proposed approach.

For the evaluation of the performance of the network, the random sub-sampling method was applied. In this method, K data splits of the overall data set were performed. Each split was randomly selected with a fixed number of examples. 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.

(2)

(4)

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

K

We experimented with the proposed network using the parameters outlined in Table 1 and Table 2 with the weight factor [8].

Table 1. Initial Parameters for GAs

Parameter	Value
Generation	100
Population size	50
Crossover rate	0.65
Mutation rate	0.1

Table 2. Initial Parameters for FCM-based FNNm

Parameter	Value
Fuzzification coefficients	$1.0<m_i$□2.5
Learning rate	$0.0\leq\eta\leq 0.01$
Moment coefficient	$0.0\leq\alpha\leq 0.001$

4.1. Iris Dataset

In this section, we use the Iris dataset [9]. 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.

Table 3 and Figure 6 show the performance of CR and PI for FCM-based FNNm before optimization. Table 4 and Figure 7 present the performance of CR and PI for FCM-based FNNm using genetic optimization. From these tables and figures we know that the optimized FCM-based FNNm is better than before optimization.

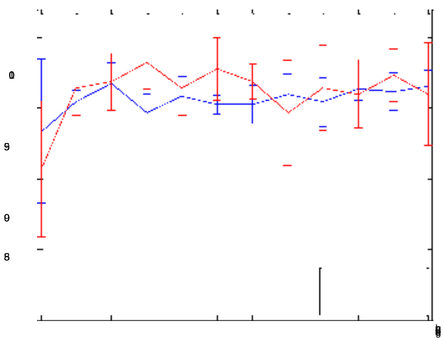
Table 3. Performance of the FCM-based FNNm for Iris Dataset

No. of Clusters	Inference (Type)	CR Training	CR Testing	PI Training	PI Testing
5	1	93.33±5.13	90.67±4.82	0.035±0.02	0.040±0.01
	2	95.43±0.80	96.44±1.99	0.026±0.00	0.028±0.01
	3	96.76±1.44	96.89±1.99	0.025±0.00	0.027±0.01
10	1	94.67±1.28	98.22±1.86	0.024±0.00	0.022±0.01
	2	95.81±1.44	96.44±1.99	0.019±0.00	0.025±0.01
	3	95.24±0.67	97.78±2.22	0.022±0.00	0.024±0.01
15	1	95.24±1.35	96.89±1.22	0.023±0.00	0.023±0.01
	2	96.00±1.41	94.67±3.72	0.022±0.00	0.023±0.01
	3	95.43±1.70	96.44±2.98	0.021±0.00	0.022±0.01
20	1	96.38±0.80	96.00±2.43	0.021±0.00	0.026±0.01
	2	96.19±1.35	97.33±1.86	0.023±0.00	0.023±0.01
	3	96.57±1.09	96.00±3.65	0.022±0.00	0.024±0.01

Table 4. Performance of the Optimized FCM-based FNNm for Iris Dataset

No. of Clusters	Inference (Type)	CR Training	CR Testing	PI Training	PI Testing
5	1	96.95±1.56	96.89±3.72	0.022±0.01	0.027±0.01
	2	99.05±0.67	97.78±1.57	0.029±0.00	0.033±0.01
	3	98.86±0.43	97.78±1.57	0.023±0.00	0.024±0.00
10	1	96.76±0.52	99.56±0.99	0.022±0.00	0.011±0.01
	2	98.29±0.43	99.56±0.99	0.031±0.00	0.026±0.00
	3	98.67±0.52	98.67±1.22	0.026±0.00	0.023±0.00
15	1	98.67±0.85	96.89±2.53	0.013±0.01	0.020±0.02
	2	99.05±0.67	97.78±1.57	0.021±0.01	0.026±0.01
	3	99.05±0.00	97.78±0.00	0.021±0.01	0.027±0.00
20	1	98.67±0.85	97.33±1.86	0.034±0.04	0.034±0.04
	2	98.67±0.85	99.56±0.99	0.014±0.01	0.011±0.01
	3	99.62±0.52	97.78±1.57	0.009±0.01	0.022±0.01

From Table 4 and Figure 7 we select the network that has twenty fuzzy rules and linear inference (Type 2) engine. This network exhibits CR=98.67±0.85, PI=0.014±0.01 for training datasets and CR=99.56±0.99, PI=0.011±0.01for testing datasets.


(a) CR

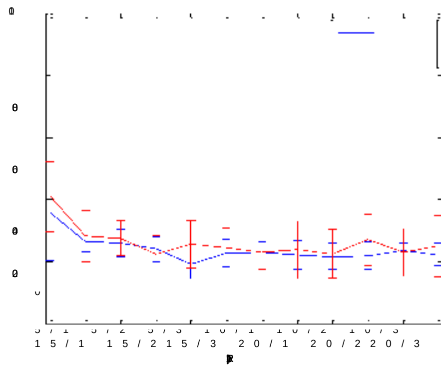
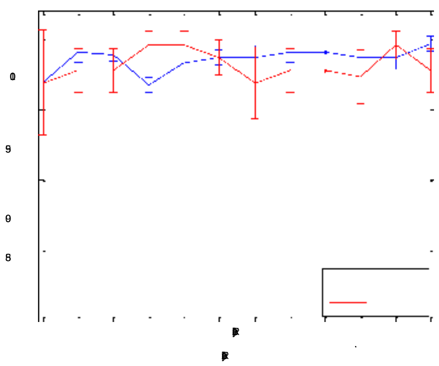

(b) PI

Figure 6. Performance of the FCM-based FNNm for Iris Dataset


(a) CR

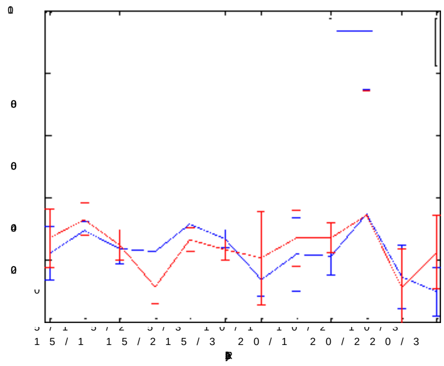

(b) PI

Figure 7. Performance of the Optimized FCM-based FNNm for Iris Dataset

Table 5 shows the confusion matrix for the selected network. The result indicates some misclassification for Versicolor and Virginica for training datasets and Virginica for testing datasets.

Table 5. Confusion Matrix for the Selected Network

(a) Training datasets

	Setosa	Versicolor	Virginica
Setosa	100.00±0.00	0.00±0.00	0.00±0.00
Versicolor	0.00±0.00	97.14±2.86	2.86±2.86
Virginica	0.00±0.00	1.14±1.56	98.86±1.56

(b) Testing datasets

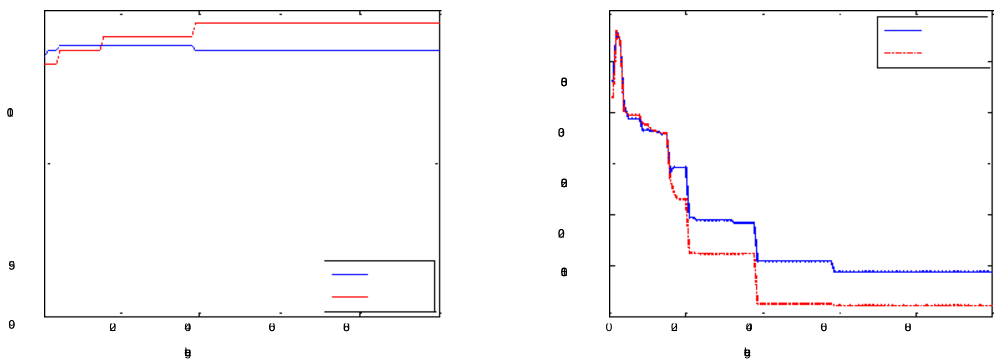
	Setosa	Versicolor	Virginica
Setosa	100.00±0.00	0.00±0.00	0.00±0.00
Versicolor	0.00±0.00	100.00±0.00	0.00±0.00
Virginica	0.00±0.00	1.33±2.98	98.67±2.98

Table 6 shows the optimized parameters of fuzzification coefficients associated with the corresponding clusters, the learning rate, and the momentum coefficient for the selected network using genetic algorithms.

Table 6. Optimized Parameters for the Selected Network

m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
1.565712	1.488977	1.382376	1.142042	2.238865	1.916492	2.176308	1.301495
m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}	m_{16}
1.928584	2.383059	1.208994	2.343709	1.662824	1.412027	1.253087	2.29295
m_{17}	m_{18}	m_{19}	m_{20}	η	α		
2.430314	1.71756	2.407883	1.887396	0.009332	0.000449		

Figure 8 presents the optimization procedure for the CR and PI when using twenty rules with Type 2 (Linear Inference) obtained in successive generations of the genetic optimization. These figures depict the average values using the random sub-sampling.



(a) CR (b) PI

Figure 8. Optimization Process for the Selected Network

The performance of the proposed model is compared with the performance of some other models reported in the literature; refer to Table 7. The comparison shows that the proposed model outperforms several previous developed models.

Table 7. Comparison of Performance with Previous Models

Model	Classification Ratio (%)
NEFCLASS [10]	96.0
C4.5 [11]	94.0
FID3.1 [12]	96.0
HNFB [13]	98.67
HNFQ [14]	98.67
HNFB-1 [15]	98.67
Our model	99.56

4.2. WDBC Dataset

In this section, we use the WDBC dataset [16]. Features of WDBC are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. The WDBC dataset is a collection of 569 instances of 2 kinds, with 30 attributes. Two classes are benign and malignant.

Here we use the simplified inference (Type I) and linear inference (Type II) to deal with the high dimensionality. Table 8 and Table 9 summarize the performances of CR and PI for FCM-based FNNm before and after optimization, respectively. Figure 9 and Figure 10 depict the performances of CR and PI in the same case.

Table 8. Performance of the FCM-based FNNm for WDBC Dataset

No. of Clusters	Inference (Type)	CR Training	CR Testing	PI Training	PI Testing
5	1	91.91±1.26	93.10±1.72	0.057±0.00	0.056±0.01
	2	97.59±0.14	96.96±1.40	0.038±0.00	0.041±0.00
10	1	92.46±0.36	91.93±0.96	0.057±0.00	0.057±0.01
	2	96.78±0.45	95.79±1.82	0.039±0.00	0.041±0.01
15	1	92.11±0.92	92.40±1.43	0.056±0.00	0.057±0.01
	2	96.23±0.50	94.97±0.98	0.038±0.00	0.043±0.00
20	1	92.41±0.84	92.87±2.20	0.054±0.00	0.057±0.01
	2	95.78±0.37	94.74±1.24	0.039±0.00	0.044±0.00

Table 9. Performance of the Optimized FCM-based FNNm for WDBC Dataset

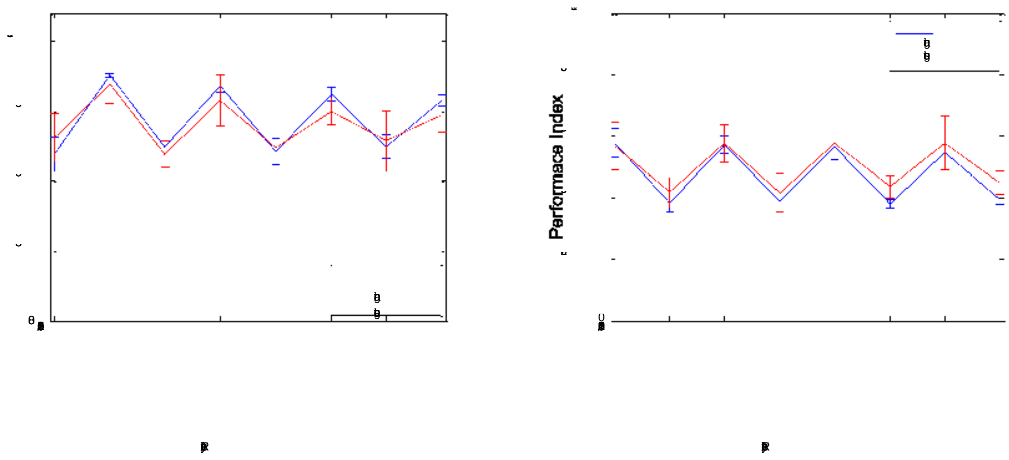
No. of Clusters	Inference (Type)	CR Training	CR Testing	PI Training	PI Testing
5	1	95.53±0.74	95.32±1.43	0.039±0.01	0.044±0.01
	2	98.59±0.38	98.13±0.96	0.041±0.01	0.042±0.01
10	1	96.03±0.45	94.85±0.76	0.035±0.00	0.042±0.00
	2	98.49±0.18	98.01±1.06	0.045±0.01	0.048±0.01
15	1	96.18±0.54	95.32±1.24	0.038±0.01	0.044±0.00
	2	99.05±0.48	97.08±1.31	0.042±0.00	0.053±0.01
20	1	96.43±0.57	95.32±1.17	0.032±0.00	0.041±0.01
	2	98.69±0.37	97.54±1.05	0.039±0.01	0.050±0.02

From Table 9 and Figure 10 we select the network that has five fuzzy rules (clusters) and linear inference (Type 2) engine. This network exhibits CR=98.59±0.38, PI=0.041±0.01 for training datasets and CR=98.13±0.96, PI=0.042±0.01 for testing dataset.

Table 10 shows the confusion matrix for the selected network. The result indicates some misclassification for both Benign and Malignant for training datasets and testing datasets.

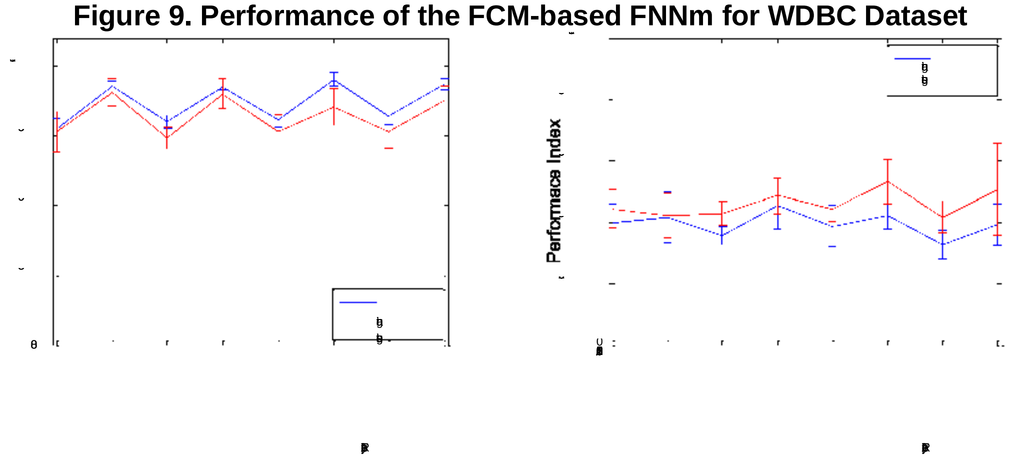
Table 11 shows the optimized parameters of fuzzification coefficients associated with the corresponding clusters, the learning rate, and the momentum coefficient for the selected network using genetic algorithms.

Figure 11 presents the optimization procedure for the CR and PI when using five rules with linear inference (Type 2) obtained by genetic optimization.



(a) CR (b) PI

Figure 9. Performance of the FCM-based FNNm for WDBC Dataset



(a) CR (b) PI

Figure 10. Performance of the Optimized FCM-based FNNm for WDBC Dataset

Table 10. Confusion Matrix for the Selected Network

(a) Training datasets

	Benign	Malignant
Benign	99.52±0.52	0.48±0.52
Malignant	2.97±1.02	97.03±1.02

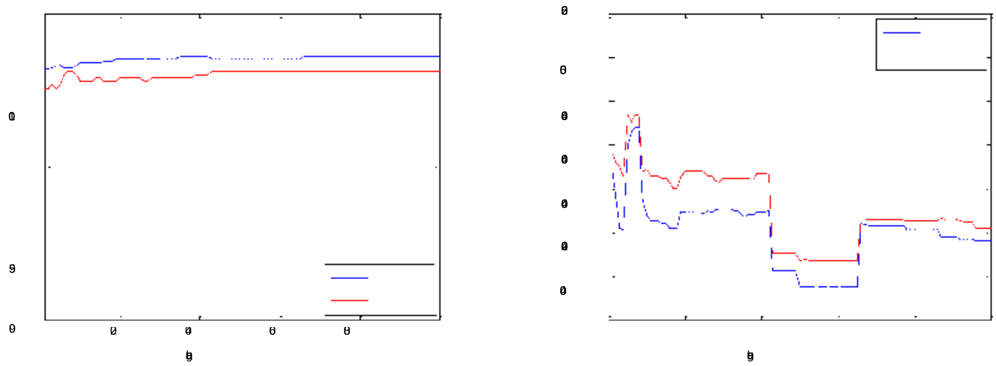
(b) Testing datasets

	Benign	Malignant
Benign	98.88±1.22	1.12±1.22
Malignant	3.13±1.56	96.88±1.56

Table 11. Optimized Parameters for the Selected Network

m_1	m_2	m_3	m_4	m_5	η	α
1.476586	1.462957	1.88349	2.335035	1.902575	0.007307	0.000259

Table 12 shows the performance of the proposed model to compare with the performance of some other models reported in the literature.



(a) CR (b) PI

Figure 11. Optimization Process for the Selected Network

Table 12. Comparison of Performance with Previous Models

Model	Classification Ratio (%)
SVM [17]	96.68
Bayes Net [18]	95.81
RVM [19]	97.2
MPANN [20]	98.1
MLP [21]	85.92
DigaNN [22]	97.9
RBF2 [23]	97.13
Our model	98.13

4.3. Wine Dataset

In this section, we use the Wine dataset [24]. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

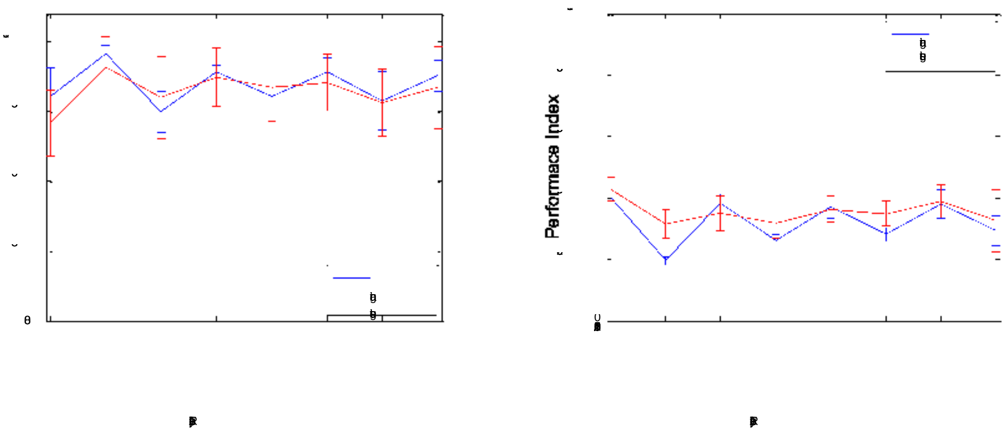
Table 13. Performance of the FCM-based FNNm for Wine Dataset

No. of Clusters	Inference (Type)	CR		PI	
		Training	Testing	Training	Testing
5	1	96.10±2.02	94.18±2.37	0.040±0.01	0.042±0.00
	2	99.19±0.57	98.18±2.23	0.019±0.00	0.031±0.00
10	1	94.96±1.45	96.00±2.99	0.038±0.00	0.035±0.01
	2	97.89±0.45	97.45±2.07	0.026±0.00	0.031±0.01
15	1	96.10±1.85	96.73±2.37	0.037±0.00	0.036±0.00
	2	95.77±2.10	95.64±2.44	0.038±0.00	0.039±0.01
20	1	97.89±0.93	97.09±2.07	0.028±0.00	0.034±0.00
	2	97.56±1.15	96.73±2.99	0.029±0.00	0.032±0.01

Here we use the simplified inference (Type I) and linear inference (Type II). Table 13 and Figure 12 show the performance of CR and PI for FCM-based FNNm before optimization. Table 14 and Figure 13 present the performance of CR and PI for FCM-based FNNm using genetic optimization.

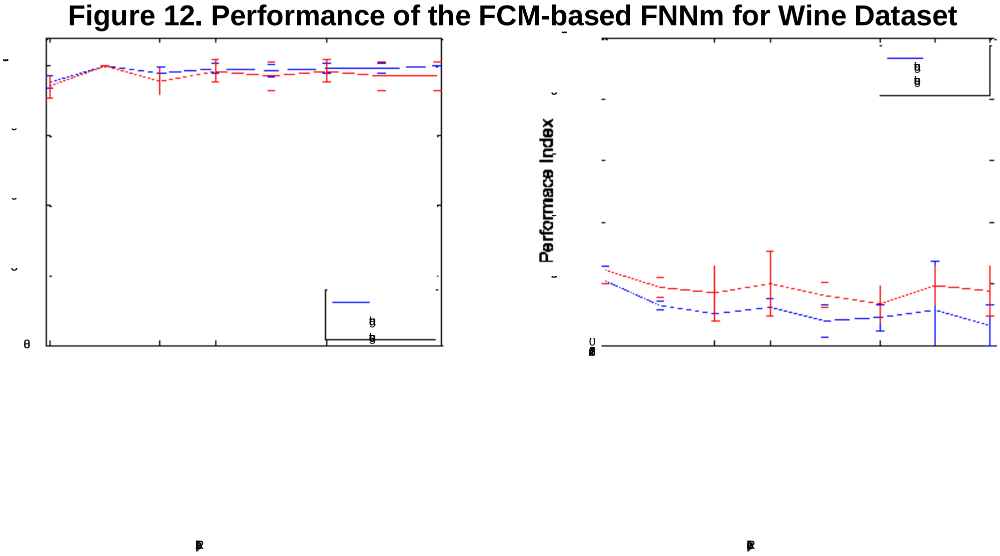
Table 14. Performance of the Optimized FCM-based FNNm for Wine Dataset

No. of Clusters	Inference (Type)	CR Training	CR Testing	PI Training	PI Testing
5	1	98.86±0.45	98.55±0.81	0.020±0.00	0.024±0.00
	2	100.00±0.00	100.00±0.00	0.012±0.00	0.018±0.00
10	1	99.51±0.45	98.91±1.00	0.010±0.00	0.016±0.01
	2	99.84±0.36	99.64±0.81	0.012±0.00	0.020±0.01
15	1	99.67±0.45	99.27±1.00	0.007±0.01	0.016±0.00
	2	99.84±0.36	99.64±0.81	0.008±0.00	0.013±0.01
20	1	99.84±0.36	99.27±1.00	0.011±0.02	0.019±0.01
	2	100.00±0.00	99.27±1.00	0.006±0.01	0.017±0.01



(a) CR (b) PI

Figure 12. Performance of the FCM-based FNNm for Wine Dataset



(a) CR (b) PI

Figure 13. Performance of the Optimized FCM-based FNNm for Wine Dataset

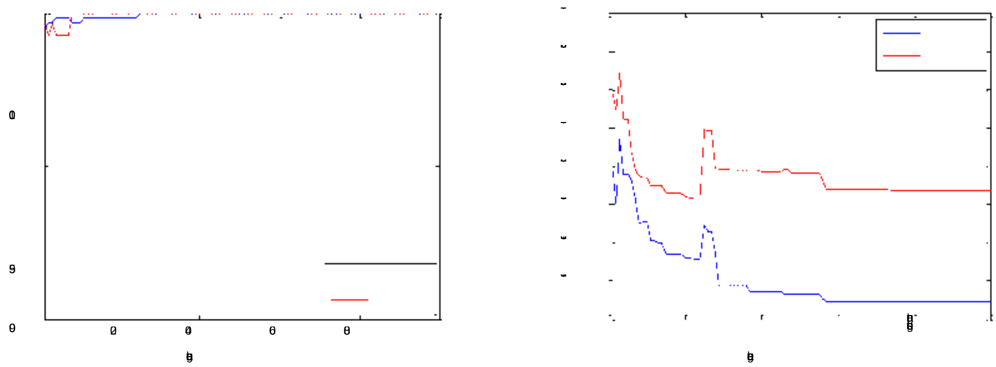
From Table 14 and Figure 13 we select the network that has five fuzzy rules (clusters) and linear inference (Type 2) engine. This network exhibits CR=100.00±0.00, PI=0.012±0.00 for training datasets and CR=100.00±0.00, PI=0.018±0.00 for testing dataset.

Table 15 shows the optimized parameters of fuzzification coefficients associated with the corresponding clusters, the learning rate, and the momentum coefficient for the selected network using genetic algorithms.

Table 15. Optimized Parameters for the Selected Network

m_1	m_2	m_3	m_4	m_5	η	α
1.145054	2.332643	1.25783	1.195322	1.726746	0.009304	0.000998

Figure 14 presents the optimization procedure for the CR and PI when using five rules with linear inference (Type 2) obtained by genetic optimization.



(a) CR (b) PI

Figure 14. Optimization Process for the Selected Network

0.012

5. Conclusions

In this paper, the design of the fuzzy c-means clustering-based fuzzy neural networks has been introduced and its optimization using real-coded genetic algorithms has been discussed for pattern classification.

The input spaces of the proposed networks were divided as the scatter form using FCM clustering algorithm to generate the fuzzy rules. The partitioned spaces describe the fuzzy rules and the number of the fuzzy rules is equal to the number of clusters. From this method, we could alleviate the problem of the curse of dimensionality and have designed fuzzy neural networks compact and easy. Effectively partitioning input space can decrease the number of fuzzy rules and thus increase the learning speed. And genetic algorithms were also used for parametric optimization of the proposed networks.

From the results in the previous section, we were able to design good networks for pattern classification. 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

[1] T. Yamakawa, "A Neo Fuzzy Neuron and Its Application to System Identification and Prediction of the System Behavior", Proceeding of the 2nd International Conference on Fuzzy logic & Neural Networks, **(1992)** July 17-22; Lizuka, Japan.
[2] J. J. Buckley and Y. Hayashi, "Fuzzy neural networks: A survey", Fuzzy Sets Syst., vol. 66, **(1994)**.
[3] M. Beldjehem, "A Unified Granular Fuzzy-Neuro Framework for Predicting and Understanding Software Quality", International Journal of Software Engineering and Its Applications, vol. 4, no. 4, **(2010)**.
[4] D. C. S. Bisht and A. Jangid, "Discharge Modelling using Adaptive Neuro - Fuzzy Inference System", International Journal of Advanced Science and Technology, vol. 31, **(2011)**.
[5] M. R. Homaeinezhad, E. Tavakkoli and A. Ghaffari, "Discrete Wavelet-based Fuzzy Network Architecture for ECG Rhythm-Type Recognition: Feature Extraction and Clustering-Oriented Tuning of Fuzzy Inference System", International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 4, no. 3, **(2011)**.
[6] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, **(1981)**.
[7] D. E. Golderg, "Genetic Algorithm in search", Optimization & Machine Learning, Addison Wesley, **(1989)**.
[8] S. K. Oh, K. C. Yoon and H. K. Kim, "The Design of Optimal Fuzzy-Neural Networks Structure by Means of GA and an Aggregate Weighted Performance Index", Journal of Control, Automation and Systems Engineering, vol. 6, no. 3, **(2000)**.
[9] http://archive.ics.uci.edu/ml/datasets/Iris.
[10] F. Klawonn, D. Nauck and R. Kruse, "Generating rules from data by fuzzy and neuro-fuzzy methods", in Proc. Fuzzy-Neuro-Systeme, **(1995)**.
[11] J. R. Quinlan, "Improved use of continuous attributes in C4.5", J. Artif. Intell. Res., vol. 4, **(1996)**.
[12] C. Z. Janikow and M. Faifer, "Fuzzy partitioning with FID3.1", in Proc. IEEE 18th Int. Conf. North American Fuzzy Information Processing Soc., **(1999)**.
[13] F. J. de Souza, M. M. B. R. Vellasco and M. A. C. Pacheco, "Load forecasting with the hierarchical neuro-fuzzy binary space partitioning model", Int. J. Comput. Syst. Signals, vol. 3, no. 2, **(2002)**.
[14] F. J. de Souza, M. M. B. R. Vellasco and M. A. C. Pacheco, "Hierarchical neuro-fuzzy quadtree models", Fuzzy Sets Syst., vol. 130, no. 2, **(2002)**.
[15] L. B. Gonçalves, M. M. B. R. Vellasco, M. A. C. Pacheco and F. J. de Souza, "Inverted hierarchical neuro-fuzzy BSP system: A novel neuro-fuzzy model for pattern classification and rule extraction in databases", IEEE Transactions on Systems, Man & Cybernetics, Part C, vol. 26, no. 2, **(2005)**.
[16] http://archive.ics.uci.edu/ml/datasets/Wine.
[17] V. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, NewYork, **(1995)**.
[18] F. Jensen, "An Introductionto Bayesian Networks", UCLPress/Springer-Verlag, **(1996)**.
[19] M. E. Tipping, "The relevance vector machine", Advances in Neural Information Processing Systems, vol. 12, **(2000)**.
[20] H. A. Abbass, "An evolutionary artificial neural networks approach for breast cancer diagnosis", Artificial Intelligence in Medicine, vol. 25, no. 3, **(2002)**.
[21] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis", Wiley, NewYork, **(2002)**.
[22] I. Anagnostopoulos and I. Maglogiannis, "Neural network-based diagnostic and prognostic estimations in breast cancer microscopic instances", Medical & Biological Engineering & Computing Springer Publishers, vol. 44, no. 9, **(2006)**.
[23] Z. R. Young, "A novel radial basis function neural network for discriminant analysis", IEEE Transactions on Neural Networks, vol. 17, no. 3, **(2006)**.
[24] http://archive.ics.uci.edu/ml/datasets/Wine.

Authors

Keon-Jun Park

He 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 Post-doc with Electrical Information Communication Engineering, Wonkwang University, Korea. His research interests include computational intelligence, intelligence modeling and control, and pattern recognition.

Jong-Pil Lee

He received the M.S degree and the Ph.D degree in Electrical Engineering from KwangWoon University, Seoul, South Korea, in 1998 and 2004, repectively. He is currently a director with Korea Electric Institute, Chungbuk, South Korea. His research interests include electrical materials and application.

Dong-Yoon Lee

He 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, repectively. He is currently a professor with Joongbu University, Chungnam, South Korea. His research interests include security system, artificial intelligence and application.