J Electr Eng Technol Vol. 8, No. 3: 621-632, 2013
http://dx.doi.org/10.5370/JEET.2013.8.3.621

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

A Design of Dynamically Simultaneous Search GA-based Fuzzy Neural Networks: Comparative Analysis and Interpretation

Byoung-Jun Park*, Wook-Dong Kim and Sung-Kwun Oh†**

Abstract – In this paper, we introduce advanced architectures of genetically-oriented Fuzzy Neural Networks (FNNs) based on fuzzy set and fuzzy relation and discuss a comprehensive design methodology. The proposed FNNs are based on 'if-then' rule-based networks with the extended structure of the premise and the consequence parts of the fuzzy rules. We consider two types of the FNNs topologies, called here FSNN and FRNN, depending upon the usage of inputs in the premise of fuzzy rules. Three different type of polynomials function (namely, constant, linear, and quadratic) are used to construct the consequence of the rules. In order to improve the accuracy of FNNs, the structure and the parameters are optimized by making use of genetic algorithms (GAs). We enhance the search capabilities of the GAs by introducing the dynamic variants of genetic optimization. It fully exploits the processing capabilities of the FNNs by supporting their structural and parametric optimization. To evaluate the performance of the proposed FNNs, we exploit a suite of several representative numerical examples and its experimental results are compared with those reported in the previous studies.

Keywords: Fuzzy set, Fuzzy relation, Fuzzy neural networks, Genetic algorithm, Polynomial fuzzy inference, Dynamically simultaneous search

1. Introduction

Computational intelligent techniques such as artificial neural networks and fuzzy logic are popular research fields because they can deal with complex problems which are difficult to solve by classical methods [1]. The fuzzy systems that are one of the most important areas of the Fuzzy Set Theory have been successfully applied to problems in system modeling [2, 3], control [4, 5] and classification [6], and resulted in a considerable number of applications. In most cases, the key for successful modeling was the ability of fuzzy systems to effectively incorporate expert knowledge [7]. Fuzzy neural networks (FNNs) and genetic fuzzy systems (GFSs) endow methods of approximate reasoning that reside within fuzzy systems with the learning mechanisms of neural networks and evolutionary algorithms. In essence, a GFS is a fuzzy system augmented by learning supported by a genetic algorithm (GA). Genetic learning support different levels of complexity of learning starting from the simplest case of parameter optimization (parametric learning) to the situations of structural learning level of complexity of learning the rule set of a rule based system [7]. FNN combines the advantages of both fuzzy inference systems in processing granular information and uncertainty and neural networks coming with learning abilities by generating a knowledge base without the need for involving human knowledge [8]. Various methods have been proposed for identification of fuzzy "if-then" rules [9, 10]. GA has been widely used for eliciting fuzzy models owing to its ability to search for optimal solutions in high-dimensional solution spaces. GA is a global optimizer based on the concepts of natural evolution [7], however when being used in a generic form it may lead to a significant computing overhead and slow convergence caused by the need to explore a huge search space [11]. To eliminate this overhead and increase the effectiveness of the underlying optimization, we introduce dynamic search-based GA that results in a rapid convergence while narrowing down the search to a limited region of the search space. To evaluate the performance of the proposed modeling approaches, we experiment with several representative numerical examples and consider a stability measure relating to the approximation and the generalization capabilities of a topology. A comparative analysis shows that the proposed topologies come with higher accuracy and predictive capability in comparison with some other models reported in the literature.

The objective of this study is to introduce advanced topologies of genetically-oriented fuzzy set/relation neural networks (FSNNs/FRNNs) and to develop a general design methodology of the proposed FNNs modeling in the comprehensive evolutionary development environment supporting structural and parametric optimization. In order to build optimal FNNs, the structure and the parameter optimization are embedded into genetic algorithm (GA). The structure optimization involves the determination of

† Corresponding Author: Dept. of Electrical Engineering, The University of Suwon, Korea. (ohsk@suwon.ac.kr)
* Telematics & Vehicle-IT Convergence Department, IT Convergence Technology Research Laboratory, ETRI, Korea
** Dept. of Electrical Engineering, The University of Suwon, Korea
Received: December 12, 2011; Accepted: December 4, 2012

the participating input variables standing in the premise parts and the polynomial type of the consequent part of fuzzy rules. The proposed FNNs topologies developed by GA assume that each polynomial is associated with a single rule, i.e., the orders of local polynomials are different in each fuzzy rule. This expresses characteristics of local spaces being formed for nonlinear analysis. The parameters of the membership functions for each input variables in the premise part are identified by GA while a parametric refinement of the consequence part of the network is realized through standard back-propagation style of learning.

The study is organized as follows. In Section 2, we discuss a structure of the fuzzy relation neural networks based on polynomial inference and elaborate on the development of the networks. The detailed genetic design of the FNNs comes with an overall description of a detailed design methodology in Section 3. In Section 4, we report on a comprehensive set of experiments. Finally concluding remarks are covered in Section 5.

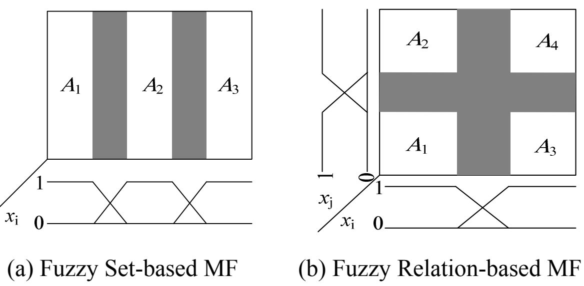
2. Architecture of the Proposed Fuzzy Neural Networks

In In this section, we discuss the concept and the algorithmic details of the proposed fuzzy neural networks (FNNs) based on polynomial inference.

2.1 Fuzzy set neural networks

The fuzzy partitioning of the spaces of input variables is divided into two types as shown in Fig.1. In the fuzzy set-based method, we are concerned with a granulation carried out in terms of fuzzy sets defined in each input variable. In the fuzzy relation-based method, Space partitioning is realized in terms of all input variables being considered simultaneously. We also use triangular membership functions as illustrated in Fig. 1.

For an efficient partition, the fuzzy partitioning defined in premise part of rules should be focused on essential regions in the data meaning that all data should be adequately "covered" so that the model is able to infer an output for any input. The quality of the overall fuzzy model heavily depends on the fuzzy partitioning which underlines a need to a careful determination of fuzzy sets as well as a suitable selection of the input variables. We will deal with these problems within the GA optimization framework.



(a) Fuzzy Set-based MF (b) Fuzzy Relation-based MF

Fig. 1. Fuzzy partitioning of input variables

As visualized in Fig. 1(a), fuzzy set-based neural networks (FSNNs) can be designed by using space partitioning realized in terms of the individual variables. The fuzzy partitions formed for the individual variables gives rise to the topology visualized in Fig. 2. The output $f_k(x_k)$ of the "∑" neuron is described as a non-linear function fi as shown in Fig. 2. In this sense, we can regard each f_i as the following mappings (rules).

$$R^i : If\ x_k\ is\ A_{ki}\ then\ Cy_{ki} \tag{1}$$

$$\begin{cases} \text{(Type0) Zero-order}: Cy_{ki} = w_{ki} \\ \text{(Type1) First-order}: Cy_{ki} = w_{ki} + w_{ki}x_k \\ \text{(type2) Second-order}: Cy_{ki} = w_{ki} + w_{ki}x_k + w_{ki}x_k^2 \end{cases} \tag{2}$$

In other words, the neurons located between each input variable and output (f_k) are represented by (2) with a nonfuzzy (numeric) conclusion, Cy_{ki}. Being more specific, R^i is the *i*-th fuzzy rule while Aki denotes a fuzzy variable of the premise of the corresponding fuzzy rule. In this paper, we propose the fuzzy polynomial inference based FNNs with respect to the form assumed by Cy_{ki}.

The topology of the proposed FSNNs is constructed by combining the scheme of fuzzy polynomial inference with neural networks. In this section, we elaborate on the pertinent algorithmic details of the design by exploiting the functionality of the individual layers of the network. The notation used in the Fig. 2 requires some clarification. The circles denote units of the FSNNs, the symbol 'N' points at a normalization procedure applied to the membership grades of the input variable x_i and '∏' refers to the product of all incoming signals. *i* is the *i*-th rule, *k* is the *k*-th input variable and w^{T0}, w^{T1}, and w^{T2} are the corresponding connections (weights).

The output of each node in the 5[th] layer is inferred by the

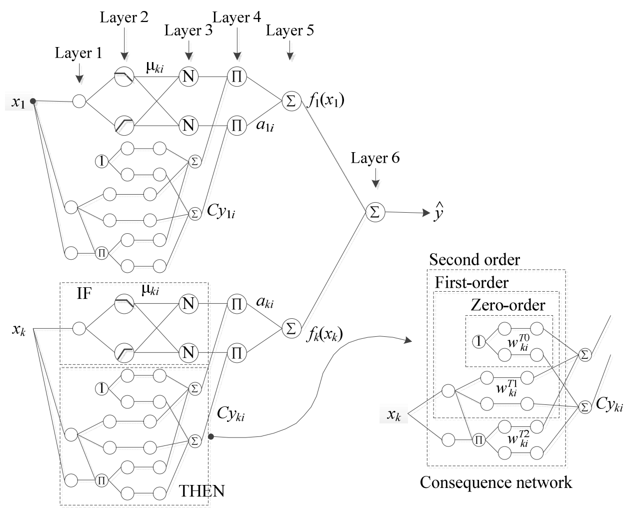


Fig. 2. Architecture of fuzzy set-based neural networks

center of gravity method expressed as

$$f_k(x_k) = \frac{\sum_{i=1}^{n} \mu_{ki} \cdot Cy_{ki}}{\sum_{i=1}^{n} \mu_{ki}} \tag{3}$$

where n is the number of membership function specified for each input variable.

The output of the FSNNs $\hat{y}$ is governed by the following expression

$$\hat{y} = f_1(x_1) + f_2(x_2) + \cdots + f_m(x_m) = \sum_{k=1}^{m} f_k(x_k) \tag{4}$$

with m being the number of the input variables (viz. the number of the outputs f_k's of the "∑" neurons in the network).

We are concerned with the number of variables in the rules and the order of the polynomial standing in the corresponding rule. The number of rules relates to the number of the membership function. The number of the fuzzy sets induces the number of the rules and the fuzzy partitions are the realizations of the linguistic terms such as LOW, HIGH, etc. Each triangular membership function A_{ki} standing in the premise part of the rule complements the neighboring (adjacent) fuzzy sets (where the complementation is sought in terms the negation of fuzzy sets). The determination of the consequence network of rules relates to the order of polynomials standing in this part of the rules. The degrees of local polynomials could vary from rule to rule. The flexibility of this nature is instrumental in nonlinear modeling completed by rule-based fuzzy models.

2.2 Fuzzy relation neural networks

In the fuzzy relation-based neural networks (FRNNs), the premise part of rules is designed by using space partitioning realized in terms of the all input variables as shown in Fig. 1(b). Fig. 3 shows the architecture of the FRNNs in case of two inputs and single output structure, where for each input we consider two membership functions

For the k-dimensional input space, the proposed FRNNs involve the following rules.

$$R^i : If\ x_1\ is\ A_{1i}\ and\ x_2\ is\ A_{2i}\ and \cdots \\ and\ x_k\ is\ A_{ki}\ then\ Cy_i \tag{5}$$

$$\begin{cases} \text{Type 0}: Cy_i = w_{0i} \\ \text{Type 1}: Cy_i = w_{0i} + w_{1i}x_1 + \cdots + w_{ki}x_k \\ \text{Type 2}: Cy_i = w_{0i} + w_{1i}x_1 + \cdots + w_{ki}x_k + w_{k+1i}x_1x_2 + \cdots + w_{k+ji}x_{k'}x_k \end{cases} \tag{6}$$

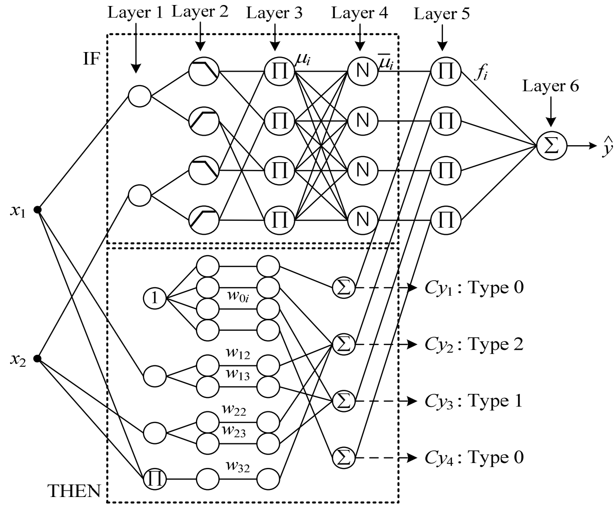


Fig. 3. Architecture of fuzzy relation-based neural networks

where, i is the i-th rule, k is the k-th input variable and w_{0i}, w_{ki}, and w_{k+ji} are connections (weights) as shown in Fig. 3. The notation used in the figure requires some clarification. The 'circles' denote units of the FRNNs, 'N' refers to a normalization procedure applied to the membership grades and '∏' and '∑' are the product and the summation operations of all incoming signals, respectively.

The polynomial of the consequence part in FRNNs leads to the networks with the connections shown in Fig. 3. This network involves simplified (Type 0), linear (Type 1) and modified quadratic (Type 2) fuzzy inference mechanism. The output $\hat{y}$ of FRNNs is determined as follows

$$\hat{y} = \sum_{i=1}^{n} f_i(\mathbf{x}) = \sum_{i=1}^{n} \bar{\mu}_i \cdot Cy_i = \sum_{i=1}^{n} \frac{\mu_i \cdot Cy_i}{\sum_{i=1}^{n} \mu_i} \tag{7}$$

The fuzzy rules of the FRNNs in (7) are constructed based on all combinations of Aki, that is, each membership function is not independent from the corresponding fuzzy inference rule. As we can observe in Fig. 3, the number of input variables and their partition realized by membership functions for the input variable may not always be equal to each other. This implies that we can produce a rather reasonable partition of space for each input variable. In other words, the partitions of the input space could be constructed by considering some relationships between inputs and output variables.

In order to avoid the rapid increase in the number of the parameters of the polynomial, we design the FRNNs comprising consequence network with various types of fuzzy rules. That is, FRNNs consists of an aggregate of a fuzzy rule composed of premise (if) and consequence (then) with a polynomial different from a polynomial type of other fuzzy rules such as (8) and Fig. 3. The polynomials of fuzzy rules result from how we look at a fuzzy subspace (a fuzzy rule) and then increase the order of the polynomial of the fuzzy rule (subspace). This methodology can help effectively reduce the number of parameters and improve the performance of the model.

$$\begin{aligned} &R^1: \text{If } x_1 \text{ is } A_{11} \text{ and } x_2 \text{ is } A_{21} \text{ then } Cy_1 = w_{01} \\ &R^2: \text{If } x_1 \text{ is } A_{11} \text{ and } x_2 \text{ is } A_{22} \text{ then } Cy_2 \\ &\quad = w_{02} + w_{12}x_1 + w_{22}x_2 + w_{32}x_1x_2 \\ &R^3: \text{If } x_1 \text{ is } A_{12} \text{ and } x_2 \text{ is } A_{21} \text{ then } Cy_3 \\ &\quad = w_{03} + w_{13}x_1 + w_{23}x_2 \\ &R^4: \text{If } x_1 \text{ is } A_{12} \text{ and } x_2 \text{ is } A_{22} \text{ then } Cy_4 = w_{04} \end{aligned} \tag{8}$$

The determination of the input variables from the set of all variables is realized independently for the premise and consequence parts of the FRNNs. Even though a variable is not selected in the premise part of FRNNs, it could appear in the consequence part of the FRNNs. This means that the variable already being eliminated from the premise part can still play an important role in the formulation of the fuzzy subspace. Therefore, such selection strategy may improve the performance of the proposed FRNNs.

2.3 Learning of fuzzy neural networks

The main advantage of fuzzy neural networks (FNNs) is that it considers both local and global relationships between inputs and outputs in system modeling. FNN can be constructed solely from the experimental process, i.e., learning process is based on input-output data. A properly trained FNN could result in significant generalization capabilities.

In this paper, there are two main facets of learning of FNNs are considered, i.e., structure optimization and parameter estimation. The structural learning of the FNNs is to select input variables in premise and the polynomial type of the consequent of the fuzzy rules. The parameter learning of the proposed networks involves the location of membership functions in the premise and connection weights (coefficients of polynomial) in the consequence of fuzzy rules. The development of the proposed network is realized through genetic optimization (GA) and back propagation (BP).

We deal with the approach to build appropriate structure of FNNs and to determine membership parameters in the premise part of fuzzy rules using genetic algorithm while the parameters in consequence of rules are adjusted by the standard BP method based on the observed input and output data. This helps address important issues of structural optimization and reaching a global minimum when carrying out an extensive parametric optimization.

As mentioned above, the parametric refinement of the consequence part of the FNNs is realized by adjusting the values of the connections see Fig. 2 and 3, through the standard BP algorithm. We use the squared error measure to quantify the learning error of the form

$$E_p = (y_p - \hat{y}_p)^2 \tag{9}$$

where E_p is an error for the *p-th* data, y_p is the *p-th* target output data and $\hat{y}_p$ stands for the *p-th* output of the network for the corresponding data. For N input-output data pairs, an overall (global) performance index is a sum of errors

$$E = \frac{1}{N}\sum_{p=1}^{N}(y_p - \hat{y}_p)^2 \tag{10}$$

As far as learning is concerned, the connections are adjusted while the updates are quite standard

$$w(new) = w(old) + \Delta w, \text{ where } \Delta w = \eta\left(-\frac{\partial E_p}{\partial w}\right) \tag{11}$$

For the parameters in the consequent of the FNNs shown in Fig. 2, we derive the changes Δw as follows

In case of FSNNs:

$$\begin{aligned} &\Delta w_{ki}(t+1) = 2\eta(y_p - \hat{y}_p)\bar{\mu}_{ki} + \alpha\Delta w_{ki}(t) && \text{Type 0} \\ &\Delta w_{ki}(t+1) = 2\eta(y_p - \hat{y}_p)\bar{\mu}_{ki}x_i + \alpha\Delta w_{ki}(t) && \text{Type 1 (with Type 0)} \\ &\Delta w_{ki}(t+1) = 2\eta(y_p - \hat{y}_p)\bar{\mu}_{ki}x_i^2 + \alpha\Delta w_{ki}(t) && \text{Type 2 (with Type 0, 1)} \end{aligned} \tag{12}$$

In case of FRNNs:

$$\begin{aligned} &\Delta w_{0i}(t+1) = 2\eta(y_p - \hat{y}_p)\bar{\mu}_{i} + \alpha\Delta w_{0i}(t) && \text{Type 0} \\ &\Delta w_{ki}(t+1) = 2\eta(y_p - \hat{y}_p)\bar{\mu}_{i}x_i + \alpha\Delta w_{ki}(t) && \text{Type 1 (with Type 0)} \\ &\Delta w_{k+ji}(t+1) = 2\eta(y_p - \hat{y}_p)\bar{\mu}_{i}x_{k'}x_k + \alpha\Delta w_{k+ji}(t) && \text{Type 2 (with Type 0, 1)} \end{aligned} \tag{13}$$

where, η and α are learning coefficients with the values constrained to the unit interval. Quite commonly to accelerate convergence, a momentum term is being added to this learning expression. In the learning process of the consequent part of the FNNs, the adjustments of weights in Second-order embrace those present in Zero-order and First-order, in the same manner the networks architecture.

The performance of the FNNs depends on the learning algorithm. The BP is commonly used and it has been proven to exhibit superior performance over other types of learning algorithms for specific problems. However, because of gradient descent, the BP method has been criticized for its shortcoming of being stuck in the local minima and sensitivity to the initial values. In order to enhance the performance of the learning process, the genetic learning is sought as a sound alternative.

3. Genetically-oriented Fuzzy Neural Networks

In this section, we introduce new architectures and comprehensive design methodologies of genetically-oriented fuzzy set neural networks. For this genetically-oriented architecture, the dynamic search-based genetic algorithm (GA) is proposed.

3.1 Dynamic genetic optimization

To carry out an efficient genetic search, one should be aware of the fact that GA could run into some pitfalls and the search could become stagnant. One factor leading to such stagnation could be an improper representation of the search space. Generally, the search space is predefined for a given system and subsequently a length of the string is fixed in advance. We introduce a dynamic search-based GA.

This version of GA determines an optimal solution through successive adjustments of the search range. The adjustments of the range are based on the moving distance of a basis solution. By a basis solution we mean the one that has been previously determined for sufficiently large space. The procedure of adjusting space in each step of the dynamic search-based GA, refer to Fig. 4, can be explained as follows.

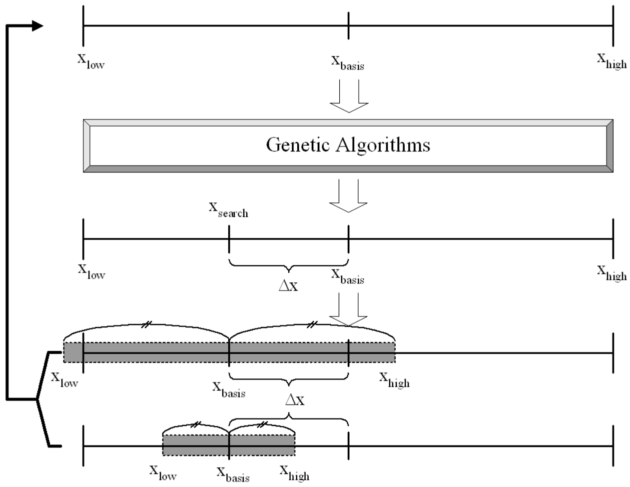


Fig. 4. Concept of the dynamic search-based GA

[Step 1] For a given problem, we set search space (range), string length for the space and a basis solution. The strings (chromosomes) are of some fixed length equal to L.

[Step 2] The genetic optimization is realized using the standard operators (reproduction, crossover, mutation).

[Step 3] Once the search has been completed meaning that the solution has been unchanged over some generations, the value is compared with the previously defined basis solution. The basis solution is reset by the solution obtained from GA, and then the search range is adjusted based on the reset basis solution to facilitate drawing an optimal solution. The movement distance (variation) of the basis solution is expressed as the difference between previously basis value and the presently selected value that is

$$\Delta x = x_{basis} - x_{search} \tag{14}$$

$$x_{basis} = x_{search} \tag{15}$$

$$x_{range} = [x_{basis} - (\varepsilon \cdot \Delta x), \quad x_{basis} + (\varepsilon \cdot \Delta x)] \tag{16}$$

where, x_{basis} is a basis solution, x_{search} result from GA, Δx is the variation (difference) between a basis solution and the selected solution, x_{range} forms a new search space and ε is an arbitrary positive constant namely, ε>0. The variation represents the movement of the basis solution.

[Step 4] For the new range, GA carries out search process by repeating Steps 2 and 3. Here, the variation of the basis solution is continuously checked, the selection of ε which we could refer to as the "variation factor" is going to be proportional to this variation. If the variation assumes large values then ε becomes equal to 1 or even larger than 1. The large variation indicates that the search range was inappropriate, so the search has to be confined to another search range such as the one illustrated in Fig. 4. Small variations indicate that the parameter is located within a neighborhood of the optimal solution, so, the search space could be further scaled down to allow for further fine tuning. While not changing the length of the string, we have accomplished an effect of the higher resolution and in this way supported the fine-grain search.

[Step 5] Once we have achieved satisfactory solution or No. further improvement has been reported, we terminate the optimization process.

The key features of the proposed dynamic search- based GA can be outlined as follows:

- It assures us that a suitable search space has been formed
- It diminishes the charge in the string length for the refinement of an optimal solution in a search space; there is No. increase in string length for refining an optimal solution.
- It increases the precision of search (viz. a precise solution could be reached without resorting to longer binary strings)

3.2 Structural and parametric optimization

As stated earlier, for optimal FNNs the structure and the parameters of the networks are subject to the GA optimization. The structure optimization involves the determination of the participating input variables in the premise and consequence parts, respectively, and the specification of the type of consequent polynomial of the fuzzy rules. Structure optimization at the level of the premise can be viewed as a problem of combinatorial optimization aimed at building significant rules out of a

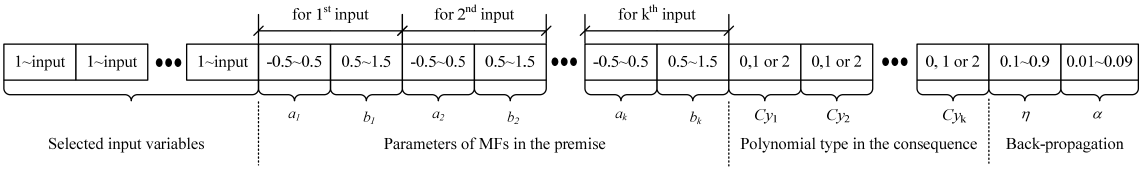


Fig. 5. Structure of chromosome for simultaneous turning optimization

given suite of input variables. The optimization of the membership parameters in the premise part concentrates on essential regions implied by the structure (distribution) of the data. All of these components of the model are constructed by running the genetic optimization.

[Step 1] We select model's inputs on a basis of the content of the chromosome. These selected inputs are used for the premise and the consequence part of the rules as illustrated in Fig. 2 and 3.

[Step 2] For the selected input variables, a subspace is defined in terms of the fuzzy relation. In order to partition the required space, the chromosome contains the values of the parameters and specifies the number of the membership functions for each input variable.

[Step 3] Here the consequence part of FNNs is decided upon on a basis of the content of the chromosome. Namely, the inputs are selected and the orders of the polynomials in the consequence part of the rules are determined.

[Step 4] The outputs of FNNs are computed.

[Step 5] In order to adjust the values of the connections, the parametric learning is invoked with the parameters η and α whose values are contained in the corresponding entries of the chromosome.

[Step 6] Each chromosome for the individual is evaluated on a basis of the fitness function computed for the optimized FNNs. Here PI is the performance index of the model whose value is determined for a given dataset.

$$Fitness\ \ function = \frac{1}{1+PI} \tag{17}$$

[Step 7] While searching for an optimal solution, Steps 1 to 6 are repeated. The best chromosome corresponds to the optimal FNNs.

4. Experimental Studies

In this section, we report on the performance of the proposed FNNs when using a number of well-known experimental benchmarks. The values of the parameters of the GA are shown in Table 1. It is worth noting that the specific numeric values of these parameters were obtained as a result of intensive experimentation. As a matter of fact, these specific numeric values are very much in line with the values that could be encountered in the literature.

Table 1. Numeric values of the GA optimization

Parameters	Values
Number of generations	1000
Population size	60
Selection operator	Tournament
Crossover operator [rate]	Two-point [0.75]
Mutation operator [rate]	Uniform [0.065]

4.1 Three-input nonlinear function

The In this experiment, we use the same numerical data as considered in [2]. The three-input nonlinear function is given as

$$y = (1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5})^2 \tag{18}$$

We consider 40 pairs of the original input-output data sampled from the input range [1, 5]. The choice of data is motivated by the possibility of running comparative analysis with some previous studies. 20 out of 40 pairs of input-output data are used as a learning set and the remaining part serves as a testing set. The performance index (PI) for the model is defined in the form

$$PI = \frac{1}{N}\sum_{p=1}^{N}\frac{|y_p - \hat{y}_p|}{y_p}\times 100 \tag{19}$$

where PI is denoted by API and GPI for the training set and the testing set, respectively.

Table 2 summarizes the results of learning. In case of (b),

Table 2. Performance index of FNNs for nonlinear function

Str	Case	Inputs	MFs	Types	API	GPI
FR	(a)-I	x_1,x_2	{2,2}	All is 0	11.73	23.70
		x_1,x_2	{2,2}	All is 1	11.52	14.78
		x_1,x_2	{2,2}	All is 2	11.01	13.93
	(b)-I	$\mathbf{x_1,x_2}$	**Tuned**	**Tuned**	**10.47**	**9.67**
	(a)-II	x_1,x_2,x_3	{2,2,2}	All is 0	1.939	5.174
		x_1,x_2,x_3	{2,2,2}	All is 1	1.705	5.168
		x_1,x_2,x_3	{2,2,2}	All is 2	11.73	23.70
	(b)-II	$\mathbf{x_1,x_2,x_3}$	**Tuned**	**Tuned**	**0.259**	**0.637**
FS	(a)-I	x_1,x_2	{2,2}	All is 0	14.94	15.10
		x_1,x_2	{2,2}	All is 1	11.14	12.93
		x_1,x_2	{2,2}	All is 2	10.99	12.76
	(b)-I	$\mathbf{x_1,x_2}$	**Tuned**	**Tuned**	**10.02**	**9.012**
	(a)-II	x_1,x_2,x_3	{3,3,3}	All is 0	6.767	9.954
		x_1,x_2,x_3	{3,3,3}	All is 1	5.082	7.188
		x_1,x_2,x_3	{3,3,3}	All is 2	4.484	7.103
	(b)-II	$\mathbf{x_1,x_2,x_3}$	**Tuned**	**Tuned**	**3.419**	**3.940**

Str: Structure; (a): Without Optimization; (b): With Optimization here the values for the 'Inputs', 'MFs' and 'Types' terms are obtained by the GA while values of (a) are selected by author in order to compare (a) with (b). In describing membership functions, the notation {2, 2, 2}, {2, 2} etc. indicates the number of membership functions assigned to the selected input variables.

As shown in Table 2, the performance of model with the optimization mechanism (b) is superior to the model without optimization (a). In case of the model without optimization (a), the values of membership function of premise part are fixed and the polynomial type of consequent part corresponding fuzzy rules is same. In case of the model with optimization (b)-II, for the selected inputs x1, x2, and x3, the performance of FRNNs is better than FSNNs.

The fuzzy rules of FR (b)-II and FS (b)-II are 2×2×2 = 8 and 3+3+3 = 9. The fuzzy rules are expressed as follows:

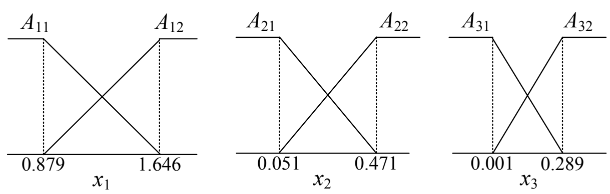
FR (b)-II:

$$\begin{aligned}
&R^1: If\ x_1\ is\ A_{11}\ and\ x_2\ is\ A_{21}\ and\ x_3\ is\ A_{31}\ then\ Cy_1 = 15.995 + 15.995x_1 + 0.79x_3 \\
&R^2: If\ x_1\ is\ A_{11}\ and\ x_2\ is\ A_{21}\ and\ x_3\ is\ A_{32}\ then\ Cy_2 = 11.038 + 12.47x_1 - 1.696x_3 - 0.683x_1x_3 \\
&R^3: If\ x_1\ is\ A_{11}\ and\ x_2\ is\ A_{22}\ and\ x_3\ is\ A_{31}\ then\ Cy_3 = 11.399 + 11.642x_1 - 0.616x_2 + 2.764x_3 \\
&\qquad 0.536x_1x_2 + 3.279x_1x_3 - 0.013x_2x_3 \\
&R^4: If\ x_1\ is\ A_{11}\ and\ x_2\ is\ A_{22}\ and\ x_3\ is\ A_{32}\ then\ Cy_4 = 7.337 + 10.870x_1 - 1.236x_2 - 0.856x_3 \\
&R^5: If\ x_1\ is\ A_{12}\ and\ x_2\ is\ A_{21}\ and\ x_3\ is\ A_{31}\ then\ Cy_5 = 2.993 \\
&R^6: If\ x_1\ is\ A_{12}\ and\ x_2\ is\ A_{21}\ and\ x_3\ is\ A_{32}\ then\ Cy_6 = 2.333 + 2.333x_1 \\
&R^7: If\ x_1\ is\ A_{12}\ and\ x_2\ is\ A_{22}\ and\ x_3\ is\ A_{31}\ then\ Cy_7 = 2.179 \\
&R^8: If\ x_1\ is\ A_{12}\ and\ x_2\ is\ A_{22}\ and\ x_3\ is\ A_{32}\ then\ Cy_8 = 2.034 + 2.034x_1 - 7.533x_2 - 4.453x_3
\end{aligned} \tag{20}$$

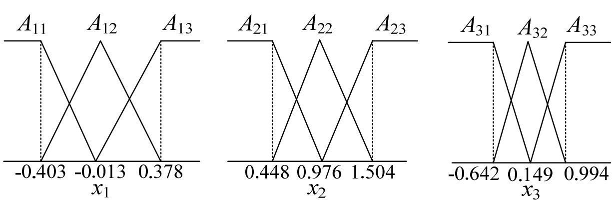
FS (b)-II:

$$\begin{aligned}
&R^1: If\ x_1\ is\ A_{11}\ \ then\ Cy_1 = 15.995 \\
&R^2: If\ x_1\ is\ A_{12}\ \ then\ Cy_2 = -7.957 \\
&R^3: If\ x_1\ is\ A_{13}\ \ then\ Cy_3 = -0.128 + 0.136x_1 + 0.136x_1^2 \\
&R^4: If\ x_2\ is\ A_{21}\ \ then\ Cy_4 = 28.787 \\
&R^5: If\ x_2\ is\ A_{22}\ \ then\ Cy_5 = -45.654 + 69.545x_2 \\
&R^6: If\ x_2\ is\ A_{23}\ \ then\ Cy_6 = 8.782 + 8.782x_2 + 8.782x_2^2 \\
&R^7: If\ x_3\ is\ A_{31}\ \ then\ Cy_7 = 15.482 \\
&R^8: If\ x_3\ is\ A_{32}\ \ then\ Cy_8 = -9.417 \\
&R^9: If\ x_3\ is\ A_{33}\ \ then\ Cy_9 = -14.151 + 3.111x_3
\end{aligned} \tag{21}$$

As shown in (20) and (21), the orders (types) of the polynomials in the consequence part of fuzzy rules are different each other. In the sequel, the consequence structure optimized by GA offers the better performance as well as its simplicity.



(a) Membership function of FRNNs



(b) Membership function of FSNNs

Fig. 6. Values of membership function of FNNs

The fuzzy rules of FR are lower than FS, however, FR consists of many parameters such as membership function and weight connection. As a result, although the fuzzy rules of FR are lower, the performance of FR is better than FS.

The estimated membership function in FRNNs and FSNNs are shown in Figs. 6(a) and (b), respectively.

Fig. 7 denotes the convergence line of performance denoted by the BP learning process.

As shown in Fig. 7, the performance index of FRNNs is API=0.259 and GPI=0.637 while FSNNs is API=3.419 and GPI=3.940. The learning rate and momentum term of FRNNs and FSNNs are 0.3996, 0.0080, 0.1351, and 10e-8, respectively.

Table 3 brings forward a comparative analysis including several previous models reported in the literature. A quadratic type of polynomial was used as a polynomial networks while a standard neural network and a RBF neural network are also used. Sugeno's model I and II were fuzzy models based on the linear fuzzy inference method while Shin-ichi's models built rules by using mechanisms of neurocomputing. The studies presented in [10] are based on fuzzy-neural networks using HCM clustering and evolutionary fuzzy granulation. Remarkably, the proposed

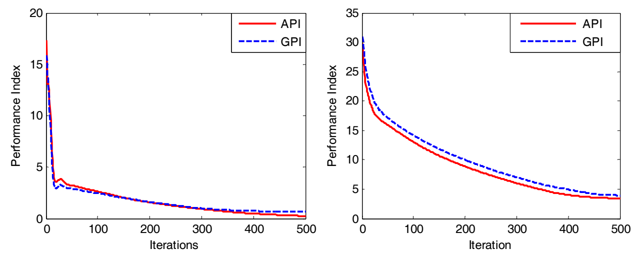


Fig. 7. BP learning process of FNNs

Table 3. Comparison analysis of performance index of several selected model

Model		*API*	*GPI*	No. of rules(nodes)
Polynomial networks		5.78	6.82	10 nodes
Standard Neural Networks		0.047	1.104	15 nodes
RBF neural networks		0.008	8.33	17 nodes
Linear model		12.7	11.1	-
GMDH [12]		4.7	5.7	-
Sugeno's [2]	Fuzzy I	1.5	2.1	3 rules
	Fuzzy II	1.1	3.6	4 rules
Shinichi's [13]	FNN Type 1	0.84	1.22	8 rules
	FNN Type 2	0.73	1.28	4 rules
	FNN Type 3	0.63	1.25	8 rules
FNN	Simplified [10]	2.865	3.206	9 rules
	Linear [10]	2.670	3.063	9 rules
Multi FNN	Simplified [14]	0.865	0.956	36 rules
	Linear [14]	0.174	0.689	18 rules
Proposed FNNs	FRNNs	0.259	0.637	8 rules
	FSNNs	3.419	3.940	9 rules

FRNNs come with higher accuracy and improved prediction capabilities.

4.2 Gas furnace time series process

Here we illustrate the performance of the network and elaborate on its development by experimenting with the classic Box-Jenkins gas furnace data. The time series data (296 input-output pairs) resulting from the gas furnace process has been intensively studied in the previous literature [19, 21].

The delayed terms of methane gas flow rate, u(t) and carbon dioxide density, y(t) are used as system input variables such as u(t-3), u(t-2), u(t-1), y(t-3), y(t-2), and y(t-1) represented as [x1,x2,x3,x4,x5,x6]. The first half of the dataset was used for training. The remaining part of the series serves as a testing set. The performance index, API and GPI, is defined as described by (10).

Table 4 shows the detailed results of FNNs. In case of the FRNNs selected three inputs (u(t-3), y(t-2), and y(t-1)), the best network came with the values of the performance index of API=0.0278 and GPI=0.0971.

Fig. 8 denotes preferred architecture obtained by GA of FRNNs and FSNNs, respectively. FRNNs and FSNNs consist of 3 inputs while fuzzy rules consist of 8 in case of FRNNs and 9 in case of FSNNs.

Table 5 denotes the values of the weight connections related the architecture of FNNs in Fig. 10.

Table 4. Values of performance index of FNNs for the gas furnace time series process

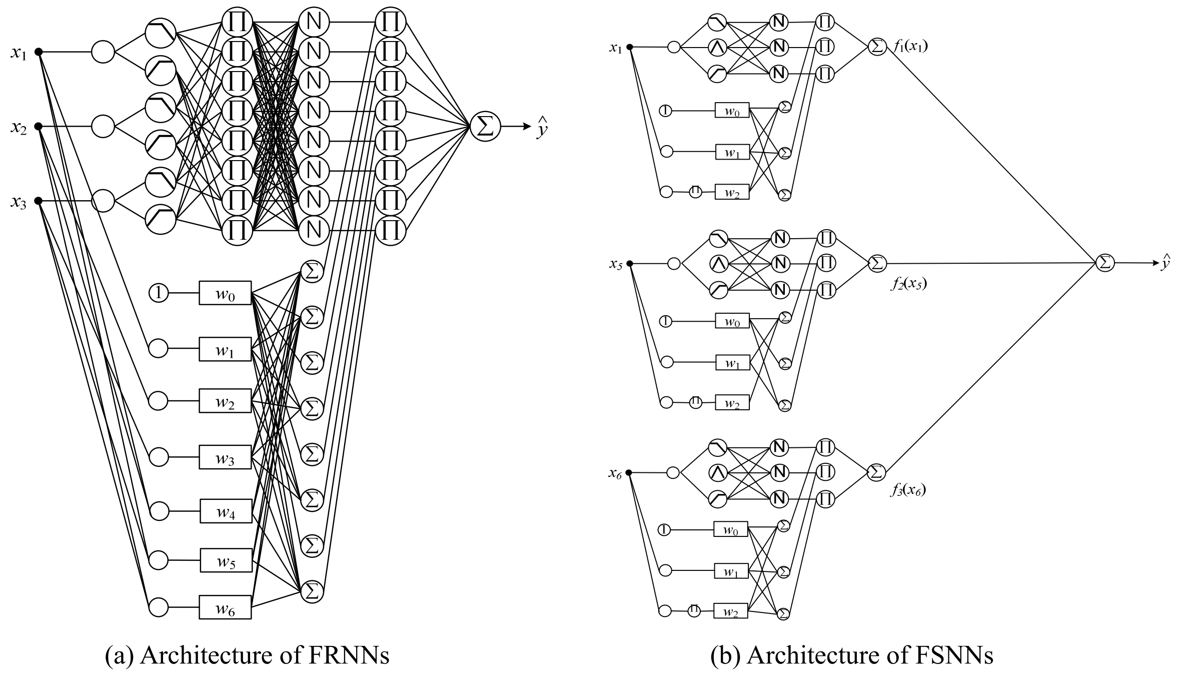
Str	Case	Inputs	MFs	Types	API	GPI
FR	(a)-I	x_1, x_6	{2,2}	All is 0	0.0234	0.3389
		x_1, x_6	{2,2}	All is 1	0.0244	0.3563
		x_1, x_6	{2,2}	All is 2	0.0245	0.3523
	(b)-I	$\mathbf{x_1, x_6}$	**Tuned**	**Tuned**	**0.0403**	**0.2674**
	(a)-II	x_1, x_5, x_6	{2,2,2}	All is 0	0.0188	0.2400
		x_1, x_5, x_6	{2,2,2}	All is 1	0.0184	0.2138
		x_1, x_5, x_6	{2,2,2}	All is 2	0.0188	0.2074
	(b)-II	$\mathbf{x_1, x_5, x_6}$	**Tuned**	**Tuned**	**0.0278**	**0.0971**
FS	(a)-I	x_1, x_6	{2,2}	All is 0	0.0231	0.3380
		x_1, x_6	{2,2}	All is 1	0.0232	0.3398
		x_1, x_6	{2,2}	All is 2	0.0236	0.3357
	(b)-I	$\mathbf{x_1, x_6}$	**Tuned**	**Tuned**	**0.0322**	**0.2938**
	(a)-II	x_1, x_5, x_6	{3,3,3}	All is 0	0.0289	0.2617
		x_1, x_5, x_6	{3,3,3}	All is 1	0.0224	02612
		x_1, x_5, x_6	{3,3,3}	All is 2	0.0223	0.2624
	(b)-II	$\mathbf{x_1, x_5, x_6}$	**Tuned**	**Tuned**	**0.0299**	**0.1073**

Str: Structure; (a): Without Optimization; (b): With Optimization

Table 5. Values of the connection weights of FNNs

S	R	Connection weights						
		w_0	w_1	w_2	w_3	w_4	w_5	w_6
FR	1	44.43	3.72	16.33	18.76	-0.99	-1.00	7.60
	2	46.46	-3.59	16.45	18.43	-4.72	-5.45	5.26
	3	37.69	-	-	-	-	-	-
	4	36.37	-1.54	19.77	8.61	-	-	-
	5	43.94	-	-	-	-	-	-
	6	62.83	-12.1	17.53	21.55	-	-	-
	7	39.26	-	-	-	-	-	-
	8	32.97	2.36	15.35	8.48	0.07	-1.45	6.97
FS	1	14.75	0.328	0.017	-	-	-	-
	2	16.22	-3.44	1.442	-	-	-	-
	3	16.15	-3.20	-	-	-	-	-
	4	20.71	2.291	0.228	-	-	-	-
	5	17.06	-	-	-	-	-	-
	6	9.359	3.158	-	-	-	-	-
	7	12.73	6.887	1.845	-	-	-	-
	8	16.05	6.792	5.523	-	-	-	-
		18.34	9.411	2.730	-	-	-	-

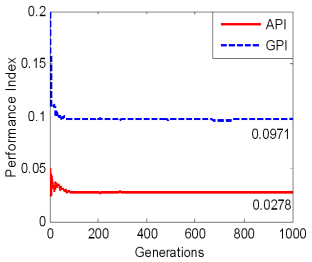
S: Structure; R: rule



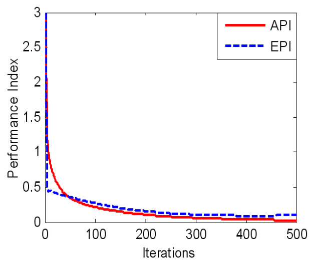
(a) Architecture of FRNNs (b) Architecture of FSNNs

Fig. 8. Architecture of the FNNs

Fig. 9 shows how the optimization was carried out by the GA and BP, respectively.

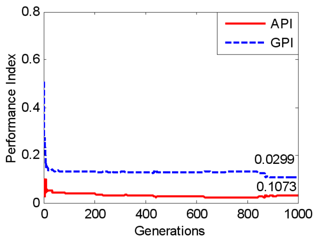


(a-1) GA optimization

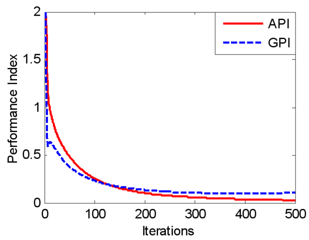


(a-2) BP learning

(a) FRNNs



(b-1) GA optimization



(b-2) BP learning

(b) FSNNs

Fig. 9. Performance index reported in successive generations and iterations

Table 6 contrasts the performance of several models with the proposed networks. The comparative results reveal that the proposed approach leads to a flexible architecture appropriate for the given dataset and shows better results than the others both in terms of better approximation and generalization capabilities and stability index.

Table 6. Comparative analysis of performance of several models of the gas furnace data

Model			*API*	*GPI*	No. of rules
Polynomial neural networks			0.022	0.143	10 nodes
Standard Neural Networks			0.021	0.398	10 nodes
RBF neural networks			0.024	0.378	15 nodes
Kim, et al.'s model [15]			0.034	0.244	2 rules
Lin and Cunningham's model [16]			0.071	0.261	4 rules
Tsekouras' model [17]			0.016	0.236	7 rules
Fuzzy	GA [18]	Simplified	0.023	0.344	4 rules
		Linear	0.018	0.264	4 rules
	HCM+GA [19]	Simplified	0.755	1.439	6 rules
		Linear	0.018	0.286	6 rules
	Hybrid [20] (GA+Complex)	Simplified	0.024	0.329	4 rules
		Linear	0.017	0.289	4 rules
Proposed FNNs		FRNNs	0.0278	0.0971	8 rules
		FSNNs	0.0299	0.1073	9 rules

4.3 NOx emission process of gas turbine plant

A NOx emission process is modeled using the data of gas turbine power plants. Till now, almost NOx emission processes are based on "standard" mathematical model in order to obtain regulation data from control process. However, such models do not efficiently capture the relationships between variables of the NOx emission process and the parameters of the model.

The input variables include AT (Ambient Temperature at site, in degrees F), CS (Compressor Speed, in rpm), LPTS (Low Pressure Turbine Speed, in rpm), CDP (Compressor Discharge Pressure, in psi), and TET (Turbine Exhaust Temperature, in degrees F). The output variable is NOx (in parts per million-volume dry). Those are denoted as 5 inputs-1 output pair [x1, x2, x3, x4, x5; y]. We consider that 130 out of 260 pairs of input-output data are used as a learning set; the remaining part of data serves as a testing set. The performance index, API and GPI, is defined as described by (10).

Table 7 summarizes the detailed results of the FNNs. The number of membership functions is equal to two, three or four for each input variable.

Table 7. Values of performance index of FNNs for the NOx emission process

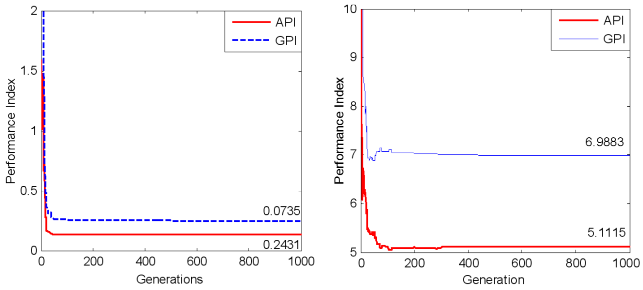
Str	Case	Inputs	MFs	Types	API	GPI
FR	(a)-I	x_2, x_4, x_5	{2,2,2}	All is 0	11.55	17.76
		x_2, x_4, x_5	{2,2,2}	All is 1	6.151	10.53
		x_2, x_4, x_5	{2,2,2}	All is 2	5.679	9.915
	(b)-I	$\boldsymbol{x_2, x_4, x_5}$	**Tuned**	**Tuned**	**1.026**	**3.381**
	(a)-II	x_2, x_3, x_4, x_5	{2,2,2,2}	All is 0	9.292	14.78
		x_2, x_3, x_4, x_5	{2,2,2,2}	All is 1	1.531	2.709
		x_2, x_3, x_4, x_5	{2,2,2,2}	All is 2	1.026	3.381
	(b)-II	$\boldsymbol{x_2, x_3, x_4, x_5}$	**Tuned**	**Tuned**	**0.130**	**0.243**
FS	(a)-I	x_3, x_4, x_5	{3,3,3}	All is 0	16.43	23.68
		x_3, x_4, x_5	{3,3,3}	All is 1	9.89	12.60
		x_3, x_4, x_5	{3,3,3}	All is 2	9.66	13.01
	(b)-I	$\boldsymbol{x_3, x_4, x_5}$	**Tuned**	**Tuned**	**4.855**	**7.072**
	(a)-II	x_2, x_3, x_4, x_5	{4,4,4,4}	All is 0	9.391	11.58
		x_2, x_3, x_4, x_5	{4,4,4,4}	All is 1	7.537	10.52
		x_2, x_3, x_4, x_5	{4,4,4,4}	All is 2	7.145	10.25
	(b)-II	$\boldsymbol{x_2, x_3, x_4, x_5}$	**Tuned**	**Tuned**	**5.111**	**6.988**

Str: Structure; (a): Without Optimization; (b): With Optimization

From the obtained results, we conclude that FRNNs model is an effective alternative, which outperforms the model formed the fuzzy set. For NOx emission process of gas turbine power plant, the FRNNs with 4 inputs and 16 fuzzy rules are preferred. It comes with the values of the performance API=0.130, and GPI=0.243. Fig. 10 shows the genetic optimization of the FRNNs and FSNNs.

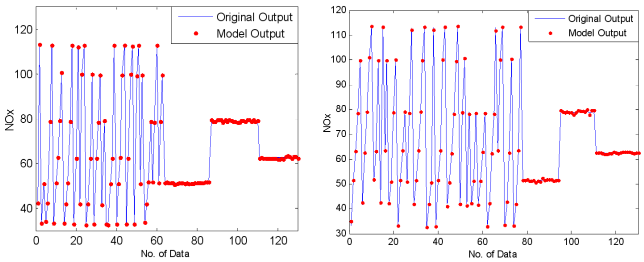
The outputs of the FRNNs for the training data and testing data are as shown in Fig. 11. We note a significant coincidence between the data and the results produced by the model.

Table 8 covers a comparative analysis including several previously built neural networks and FNN. FNN [10] was identified by hybrid algorithm combining GAs with modified complex method. The experimental results clearly reveal that the proposed approach and the resulting



(a) FRNNs (b) FSNNs

Fig. 10. GA optimization of FNNs



(a) Training (b) Testing

Fig. 11. Comparison of output of the FRNNs

Table 8. Comparative analysis of performance of several models

Model		*API*	*GPI*	No. of rules (nodes)
Polynomial networks		0.941	1.806	21 nodes
Standard Neural Networks		0.015	3.851	25 nodes
RBF neural networks		0.275	1.469	15 nodes
Regression model		17.68	19.23	-
FNN [10]	Simplified	6.269	8.778	30 rules
	Linear	3.725	5.291	30 rules
Multi FNN [14]	Simplified	2.806	5.164	120 rules
	Linear	0.720	2.025	120 rules
Proposed FNNs	FRNNs	1.026	3.381	8 rules
		0.130	0.243	16 rules

model outperform the existing networks.

4.4 Weather information of izmir

We consider the weather information of Izmir in turkey from 01/01/1994 to 31/12/1997. In order to reduce CPU time, the number of generations for GA is set to 500 times and other parameters are equal as shown in Table 2.

Table 9 shows the values of the performance index of FSNN and FRNN for weather information dataset.

Table 10 offers a comparative analysis including several models used in the literature. The multiobjective linguistic fuzzy rule-based system [21, 22] were based on the Mamdani fuzzy system and employed the multiobjective optimization techniques both concurrently learn rules and parameters and to learn only rules.

Table 9. Values of performance index of FNNs

Str	Case	Inputs	MFs	Types	API	GPI
FR	(a)-I	x_1,x_2,x_3,x_7	All is 2	0	0.832	0.946
		x_1,x_2,x_3,x_7	All is 2	1	0.825	0.973
		x_1,x_2,x_3,x_7	All is 2	2	0.832	0.983
	(b)-I	$\mathbf{x_1,x_2,x_3,x_7}$	**Tuned**	**Tuned**	**0.556**	**0.640**
FS	(a)-I	x_1,x_2,x_3,x_6,x_7,x_8	All is 2	0	1.140	1.308
		x_1,x_2,x_3,x_6,x_7,x_8	All is 2	1	0.773	1.010
		x_1,x_2,x_3,x_6,x_7,x_8	All is 2	2	1.317	1.520
	(b)-I	$\mathbf{x_1,x_2,x_3,x_6,x_7,x_8}$	**Tuned**	**Tuned**	**0.702**	**0.889**

Str: Structure; (a): Without Optimization; (b): With Optimization

Table 10. Comparative analysis of performance index of several selected models

Model		*API*	*GPI*	No. of rules (nodes)
NSGA-II Linguistic fuzzy rule-based system[21]	Rule base	2.05	2.38	23 rules
	Knowledge base	1.64	1.91	22 rules
PAES Linguistic fuzzy rule-based system[21]	Rule base	1.62	1.89	25 rules
	Knowledge base	1.30	1.49	25 rules
MOEA Linguistic fuzzy system[22]		0.92	1.10	29 rules
Proposed FNNs	FRNNs	0.56	0.64	16 rules
	FSNNs	0.73	0.72	12 rules

4.5 Economic information of USA

This dataset contains the Economic data information of USA from 01/04/1980 to 02/04/2000 on a weekly basis. This dataset includes 1049 input-output pairs. There are fifteen input variables. The number of generation of GA is set to 500 times and other parameters are equal as shown in Table 2.

Table 11. Values of performance index of FNNs

Str	Case	Inputs	MFs	Types	API	GPI
FR	(a)-I	x_5,x_7,x_{11},x_{12}	All is 2	0	0.014	0.099
		x_5,x_7x_{11},x_{12}	All is 2	1	0.010	0.082
		x_5,x_7,x_{11},x_{12}	All is 2	2	0.013	0.095
	(b)-I	$\mathbf{x_5,x_7,x_{11},x_{12}}$	**Tuned**	**Tuned**	**0.022**	**0.032**
FS	(a)-I	$x_1,x_2,x_5,x_7,x_{11},x_{13},$	All is 2	0	0.029	0.048
		$x_1,x_2,x_5,x_7,x_{11},x_{13},$	All is 2	1	0.027	0.034
		$x_1,x_2,x_5,x_7,x_{11},x_{13},$	All is 2	2	0.024	0.032
	(b)-I	$\mathbf{x_1,x_2,x_5,x_7,x_{11},x_{13}}$	**Tuned**	**Tuned**	**0.018**	**0.019**

Str: Structure; (a): Without Optimization; (b): With Optimization

Table 12. Comparison analysis of performance index of several selected model

Model		*API*	*GPI*	No. of rules (nodes)
NSGA-II Linguistic fuzzy rule-based system[21]	Rule base	0.16	0.22	7 rules
	Knowledge base	0.012	0.18	8 rules
PAES Linguistic fuzzy rule-based system[21]	Rule base	0.08	0.12	9 rules
	Knowledge base	0.08	0.14	11 rules
MOEA Linguistic fuzzy system[22]		0.04	0.05	18 rules
Multi-Genetic fuzzy system[23]		0.06	0.09	7 rules
Proposed FNNs	FRNN	0.022	0.032	16 rules
	FSNN	0.018	0.019	12 rules

Table 11 shows the performance index of FNNs for economic information dataset.

Table 12 reports on the performance of several models with the proposed model. The multi-genetic fuzzy system [31] was initialized by a method that combines the benefits of Wang-Mendel (WM) and decision-tree algorithm and then the optimization is performed by NSGA-II.

5. Concluding Remarks

In this study, we introduce advanced architectures of genetically-oriented Fuzzy Neural Networks (FNNs) based on fuzzy set and fuzzy relation. Given the large search space associated with the development of FNNs, we enhance the search capabilities of GAs by introducing their dynamic variants. The series of numeric experiments quantify the efficiency of the proposed approach and shed light on the capabilities of the designed architectures.

The proposed FNNs are based on the rule-based fuzzy neural networks with the extended structure of the premise and the consequence parts of fuzzy rules being formed within the networks. Here three different forms of regression polynomials (constant, linear and quadratic) are used in the consequence part of the fuzzy rules of FNNs. The structure and parameters of the proposed FNNs are genetically optimized.

In the FRNNs and FSNNs, we showed that the proposed fuzzy relation-based neural networks (FRNNs) can be efficiently carried out both at the structural as well as parametric level. As seems to be quite intuitive, the simultaneous optimization of the structure and parameters of the networks yields better results in comparison to the model without the aid of optimization. A suite of experimental studies shows better performance of the networks introduced in this study in comparison to the results obtained for some standard neuro-fuzzy models available in the literature

Acknowledgements

This work was supported by the GRRC program of Gyeonggi province [GRRC SUWON 2012-B2, Center for U-city Security & Surveillance Technology] and also supported by National Research Foundation of Korea Grant funded by the Korean Government (NRF-2012-003568).

References

[1] F. Herrera, "Genetic fuzzy systems: taxonomy, current research trends and prospects," Evolutionary Intelligence, Vol. 1, No. 1, pp. 27-46, 2008.

[2] G. Kang, and M. Sugeno, "Fuzzy modeling," *Trans. of the Society of Instrument and Control Engineers*, Vol. 23, No. 6, pp. 106-108, 1987.

[3] W. Pedrycz, "Fuzzy modeling: paradigms and practice", Kluwer Academic Press, Dordrecht, 1996.

[4] D. Driankov, H. Hellendoorn, and M. Reinfrank, "An introduction to fuzzy control," *Springer, Berlin*, 1993.

[5] J. Y. Zhang, and Y. D. Li, "Application of genetic algorithm in optimization of fuzzy control rules," *Proceedings of the sixth International Conference on Intelligent Systems Design and Applications* (*Jinan, China, Oct. 16-18, 2006*). *ISDA'06,* pp. 529-534, 2006.

[6] Z. Chi, H. Yan, and T. Pham, "Fuzzy algorithms: with applications to image proceeding and pattern recognition," *World Scientific, Singapore*, 1996.

[7] O, Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends," *Fuzzy Set and systems*, Vol. 141, No. 1, pp.5-31, 2004.

[8] M. Tang, C. Quek, and G. S. Ng, "GA-TSK fnn: parameters tuning of fuzzy neural network suing genetic algorithms," *Expert Systems with Applications,* Vol. 29, No. 4, pp. 769-781.

[9] M. Eftekhari, and S. D. Katebi, "Eliciting transparent fuzzy model using differential evolution," *Applied Soft Computing*, Vol. 8, No. 1, pp. 466-476, 2008.

[10] S. K. Oh, W. Pedrycz, and H. S. Park, "Hybrid identification in fuzzy neural networks," *Fuzzy Set and Systems*, Vol. 138, No. 2, pp. 399-426, 2003.

[11] *W. K. Wong, X. H. Zeng, and W. M. R. Au, "*A decision support tool for apparel coordination through integrating the knowledge-based attribute evaluation expert system and the T-S fuzzy neural network," *Expert Systems with Application*, Vol. 36, No. 2, pp. 2377-2390, 2009.

[12] T. Kondo, "Revised GMDH algorithm estimating degree of the complete polynomial," *Trans. of the Society of Instrument and Control Engineers*, Vol. 22, No. 9, pp. 928-934, 1986.

[13] S. L. Horikawa, T. Furuhashi, and Y. Uchigawa, "On fuzzy modeling using fuzzy neural networks with the back propagation algorithm," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, pp. 801-806, 1992.

[14] H. S. Park, and S. K. Oh, "Multi-FNN identification based on HCM clustering and evolutionary fuzzy granulation," *Simulation Modeling Practice and Theory*, Vol. 11, No. 7-8, pp. 627-642, 2003.

[15] E. Kim, H. Lee, M. Park, and M, Park, "A simply identified sugeno-type fuzzy model via double clustering," *Information Sciences*, Vol. 110, No. 1-2, pp. 25-39, 1998.

[16] Y. Lin, and G. A. Cunningham III, "A new approach to fuzzy-neural system modeling," *IEEE Trans. on Fuzzy Systems*, Vol. 3, No. 2, pp. 190-198, 1995.

[17] G. E. Tsekouras, "On the use of the weighted fuzzy c-means in fuzzy modeling," *Advances in Engineering*

Software, Vol. 36, No. 5, pp. 287-300, 2005.

[18] B. J. Park, S. K. Oh, and T. C. Ahn, “Optimization of fuzzy systems by means of GA and weighted factor,” *The Trans. of the Korean Institute of Electrical Engineers*, Vol. 48A, No. 6, pp. 789-799.

[19] B. J. Park, W. Pedrycz, and S. K. Oh, “Identification of fuzzy models with the aid of evolutionary data granulation,” *IEE Proceedings-Control theory and application*, Vol. 148, No. 5, pp. 406-418, 2001.

[20] S. K. Oh, W. Pedrycz, and B. J. Park, “Hybrid identification of fuzzy rule-based models,” *International Journal of Intelligent Systems*, Vol. 17, No. 1, pp. 77-10, 2002.

[21] R. Alcala, P. Ducange, F. Herrera, B. Lazzerini, and F. Marcelloni, “A multi-objective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy rule-based systems,” *IEEE Trans. Fuzzy Systems*, Vol. 17, No. 5, pp. 1106-1122, 2009.

[22] M. J. Gacto, R. Alcala, and F. Herrera, “Integration of an index to preserve the semantic interpretability in the multi-objective evolutionary rule selection and tuning of linguistic fuzzy system,” *IEEE Trans. Fuzzy Systems*, Vol. 18, No. 3, pp. 515-531, 2010.

[23] P. Pulkkinen, and H. Koivisto, “A dynamically constrained multiobjective genetic fuzzy system for regression problems,” *IEEE Trans. Fuzzy Systems*, Vol. 18, No. 1, pp. 161-177, 2010.

Byoung-Jun Park He received the B.S. and M.S. degrees in control and instrumentation engineering and Ph.D. degrees in electrical engineering from Wonkwang University, Iksan, Korea, in 1998, 2000, and 2003, respectively. He is currently a senior member of engineering staff in Telematics & Vehicle-IT Convergence Research Department, IT Convergence Technology Research Laboratory, Electronics and Telecommunications Research Institute (ETRI) in Korea. From 2005 to 2006, he held a position of a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include computational intelligence, fuzzy modeling, neurofuzzy systems, pattern recognition, granular and relational computing, and proactive idle stop.

Wook-Dong Kim He received the B.Sc., and M.Sc. degrees in electrical engineering from The University of Suwon, South Korea, in 2009, and 2011 respectively. He is currently a PhD student at electrical engineering, The University of Suwon, South Korea. His research interests include fuzzy set, neural networks, evolutionary algorithm, computational intelligence and statistical learning.

Sung-Kwun Oh He received his B.Sc., M.Sc., and Ph.D. degrees in Electrical Engineering from Yonsei University, Seoul, Korea in 1981, 1983, and 1993, respectively. From 1983 to 1989, he was a Senior Researcher of the R&D Laboratory of Lucky-Goldstar Industrial Systems Co., Ltd. From 1996 to 1997, he was a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor in the Department of Electrical Engineering, University of Suwon, Suwon, South Korea. His research interests include fuzzy system, fuzzy-neural networks, automation systems, advanced computational intelligence, and intelligent control. He also currently serves as the Associate Editor of the *KIEE Transactions on Systems and Control*, *Inter-national Journal of Fuzzy Logic* and *Intelligent Systems of the KFIS*, and *International Journal of Control, Automation, and Systems of ICASE*, South Korea.