

# Genetic Design of Independent Input Rule-Based Fuzzy Neural Networks

Keon-Jun Park<sup>1</sup>, and Yong-Kab Kim<sup>1\*</sup>

<sup>1</sup> Department of Information and Communication Engineering,  
Wonkwang University, 344-2, Shinyong-dong, Iksan-si, Chonbuk, 570-749 South Korea  
bird75@wonkwang.ac.kr, ykim@wonkwang.ac.kr

**Abstract.** In this paper, we introduce the genetic design of independent input rule-based fuzzy neural networks. The premise part of the rules of the proposed networks is realized by partitioning of the independent input space using hard-c means clustering. The independently partitioned spaces express the fuzzy rules for respective inputs. The consequence part of the rules is represented by polynomial functions. And the proposed networks are optimized using real-coded genetic algorithms to find the structure and estimate the parameters of the proposed networks. The proposed networks are evaluated the validity using numerical example for nonlinear process.

**Keywords:** Fuzzy Neural Networks (FNNs), Independent Input Rule, Genetic Algorithms (GAs), Hard C-Means (HCM) Clustering, Nonlinear Process.

## 1 Introduction

Fuzzy neural networks (FNNs) [1], [2], [3] have been mostly developed by integrating of the fuzzy model and neural networks. Typically, FNNs are represented by fuzzy “if-then” rules, while the parameters of their networks are optimized by back propagation (BP) algorithm. The generation of the fuzzy rules and the adjustment of their membership functions of the FNNs were conducted by trial and error and/or on the basis of the researcher’s experience. The designers find it difficult to develop adequate fuzzy rules and membership functions to reflect the essence of the data. Many researchers have studied and applied for the issues [4], [5], yet the problems of finding good parameters of the fuzzy sets and of partitioning spaces in the rules remain open.

In this paper, we introduce architecture of fuzzy neural networks based on independent input rule. The premise part of the rules of this network is realized by partitioning the input space using hard-c means clustering (HCM) [6] to reflect the essence of the data. The consequence part of the rules is represented by polynomial functions. The coefficients of the polynomial functions are learned by the BP algorithm. We also optimize the structure and parameters of the proposed networks

---

\* Corresponding author : ykim@wonkwang.ac.kr

using real-coded genetic algorithms (GAs) [7]. The proposed networks are evaluated through numerical experimentation for nonlinear model.

## 2 Design of the Independent Input Rule-based FNNs

The structure of the independent input rule-based FNNs (IIR-based FNNs) emerges at a junction of fuzzy sets by means of the grid partition of independent input space in the premise part and neural networks present in the consequence part of the rules. The structure of the IIR-based FNNs is composed of six-layers.

The proposed IIR-based FNNs are implied by the fuzzy grid partition of independent input space. In this sense, each rule can be viewed as the following format.

$$R^{kc} : \text{If } x_k \text{ is } A_{kc} \text{ Then } y_{kc} = f(x_1, \dots, x_d). \quad (1)$$

As far as inference schemes are concerned, we distinguish these cases:

Type 1 (Simplified Inference):

$$f = w_{kc}^0$$

Type 2 (Linear Inference):

$$f = w_{kc}^0 + \sum_{i=1}^d w_{kc}^i x_k$$

Type 3 (Modified Quadratic Inference):

$$f = w_{kc}^0 + \sum_{i=1}^d w_{kc}^i x_k + \sum_{i=1}^d \sum_{j=i+1}^d w_{kc}^z x_k x_j.$$

To be more specific,  $R^{kc}$  is the  $k$ ,  $c$ -th fuzzy rule, while  $A_{kc}$  denotes  $k$ ,  $c$ -th membership function.  $w$ 's are consequent parameters of the rules.

The functionality of each layer is described as follows.

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

[Layer 2] The nodes here calculate the membership degrees using HCM algorithm [6].

$$\mu_{kc}(x_k) = HCM(x_k, y). \quad (2)$$

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

$$\hat{f}_{kc} = \mu_{kc} / \sum_{c=1}^g \mu_{kc} = \mu_{kc}, \quad \sum_{c=1}^g \mu_{kc} = 1. \quad (3)$$

[Layer 4] The nodes in this layer compute the local output of each rule.

$$a_{kc} = \hat{f}_{kc} y_{kc}. \quad (4)$$

[Layer 5] The nodes in this layer calculate the local output of each input.

$$h_k = \sum_{c=1}^g a_{kc}. \quad (5)$$

[Layer 6] The node in this layer computes the final output.

$$\hat{y} = \sum_{k=1}^d h_k \cdot \quad (6)$$

The parametric learning of the network is realized by adjusting connections of the neurons by running a standard back-propagation algorithm.

Genetic algorithms (GAs) [7] are heuristic search algorithm that mimics the process of natural evolution which generates solutions to optimization problems using techniques inspired by natural evolution. It has been demonstrated that GAs are useful global population-based optimizers. GAs is 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).

In order to optimize the IIR-based FNNs, we optimize structural components such as the number of input variables, input variables being selected and the number of the membership functions to be used in the premise part and the type of the polynomial function occurring in the conclusion part. The parameters of the apexes of membership function in the premise part and the learning rate and the momentum coefficient occurring in the conclusion part are also optimized in a successive manner. In the identification of the network we exploit a method of successive tuning. This tuning method involves a simultaneous identification of the structure and parameters of the network that gives more weight for parametric optimization across generations.

### 3 Experimental Studies

The time series data (296 input-output pairs) coming from the gas furnace nonlinear process has been intensively studied in the previous literature [8]. The delayed terms of methane gas flow rate  $u(t)$  and carbon dioxide density  $y(t)$  are used as six input variables organized in a vector format as  $[u(t-3), u(t-2), u(t-1), y(t-3), y(t-2), y(t-1)]$ .  $y(t)$  is the output variable. The first part of the data set (consisting of 148 pairs) was used for training purposes. The remaining part of the series serves as a testing data set.

The experiments were carried out using the environment with parameters specified in Table 1.

**Table 1.** Initial parameters

	Parameter	Value
GA	Max. generation number	200
	Population size	100
	Crossover rate	0.65
	Mutation rate	0.1
	Successive tuning	Case I Case II
IIR-based FNN	Max. input number to be selected	$1 \leq k \leq 2$
	No. of MFs	$2 \leq c \leq 5$

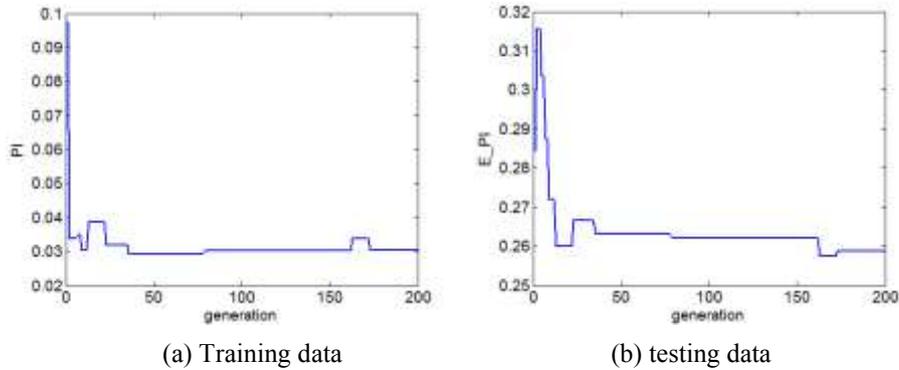
Polynomial type	$1 \leq T \leq 3$
Apexes of MFs	HCM
learning rate	$0.0 \leq \eta \leq 1.0$
moment coefficient	$0.0 \leq \alpha \leq 0.1$

Table 2 summarizes the performance for identification of the IIR-based FNN using separate method and simultaneous method. This table shows that the simultaneous method has better results than the separate method and has good generalization capabilities. From Table 2, we selected the network for Case II with 8 rules according to two input variables and the modified quadratic inference that exhibits  $PI = 0.030$ ,  $E\_PI = 0.258$ .

**Table 2.** Performance evaluation.

Identification Method		Input variables	No. of MFs	Type	PI	E_PI
Separate	Structure	1 6	5	3	0.028	0.275
	Parameters				0.030	0.264
Simultaneous	Case I	1 6	5	3	0.027	0.262
	Case II	1 6	4	3	0.030	0.258

Figure 1 depicts the values of the performance index produced in successive generations of the GAs. The model outputs of training, testing for the IIR-based FNN are presented in figure 2.



**Fig. 1.** Optimal convergence process of performance index.

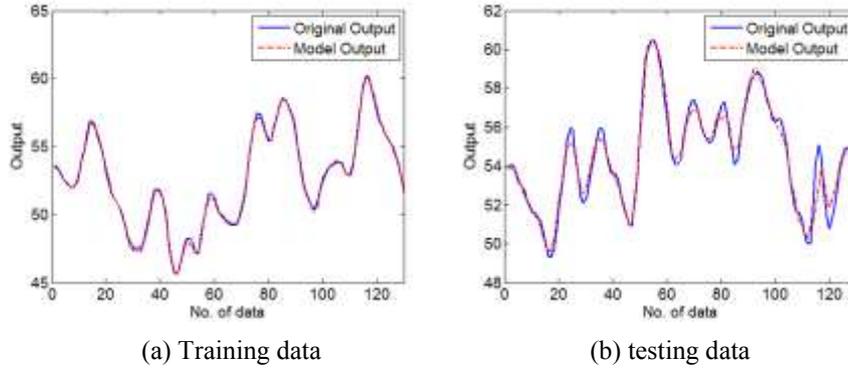


Fig. 2. Original and model outputs.

#### 4 Conclusions

This paper introduced fuzzy neural networks based on independent input space to generate the fuzzy rules and discussed its optimization using real-coded genetic algorithms with the simultaneous method. The input spaces of the proposed networks were divided as the grid form to generate the independent rules for respective inputs. And genetic algorithms were also used for the structural and parametric optimization of the proposed networks. From the results in the previous section, we were able to design good networks through the genetic design of the simultaneous method and to achieve a balance between the approximation abilities and generalization capabilities of the resulting network. Finally it could be possible to apply to many fields.

**Acknowledgements.** This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2011835).

#### References

1. Yamakawa, T.: 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) 447--483
2. Buckley, J.J., Hayashi, Y.: Fuzzy neural networks: A survey. Fuzzy Sets Syst. 66, 1--13, (1994)
3. Jang, J.S.R., Mizutani, E. and Sun, C.T.: Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence, Prentice Hall, NJ, (1997)

Proceedings, The 1st International Conference on Advanced Information and Computer Technology

4. Mateo, R. M. A., and Lee, J. W.: Intelligent Search Mechanism based on Neuro-fuzzy System for the Distributed Object Groups. *International Journal of Security and Its Applications*, 2, 1, 19--28, (2008)
5. Beldjehem, M., A Unified Granular Fuzzy-Neuro Framework for Predicting and Understanding Software Quality. *International Journal of Software Engineering and Its Applications*, 4, 4, 17--36, (2010)
6. Krishnaiah, P. R. and Kanal, L. N., editors. Classification, pattern recognition, and reduction of dimensionality, volume 2 of *Handbook of Statistics*. North-Holland, Amsterdam, (1982).
7. Golderg, D.E.: *Genetic Algorithm in search, Optimization & Machine Learning*, Addison wesley (1989)
8. Box, G.E.P., Jenkins, G.M.: *Time Series Analysis: Forecasting and Control*, 2nd ed., Holden-Day, San Francisco, CA (1976)