퍼지 및 지능시스템학회 논문지 2004, Vol. 14, No. 5, pp. 660-665

The Design of Genetically Optimized Multi-layer Fuzzy Neural Networks

Byoung-Jun Park*, Keon-Jun Park*, Dong-Yoon Lee, Sung-Kwun Oh***

*School of Electrical, Electronic and Information Engineering, Wonkwang University, Korea

*Department of Information and Communication Engineering, Joongbu University, Korea

Abstract

In this study, a new architecture and comprehensive design methodology of genetically optimized Multi-layer Fuzzy Neural Networks (gMFNN) are introduced and a series of numeric experiments are carried out. The gMFNN architecture results from a synergistic usage of the hybrid system generated by combining Fuzzy Neural Networks (FNN) with Polynomial Neural Networks (PNN). FNN contributes to the formation of the premise part of the overall network structure of the gMFNN. The consequence part of the gMFNN is designed using PNN. The optimization of the FNN is realized with the aid of a standard back-propagation learning algorithm and genetic optimization. The development of the PNN dwells on the extended Group Method of Data Handling (GMDH) method and Genetic Algorithms (GAs). To evaluate the performance of the gMFNN, the models are experimented with the use of a numerical example.

Key words : Multi-layer Fuzzy Neural Networks, Fuzzy Neural Networks, Polynomial Neural Networks, Group Method of Data Handling, Genetic Algorithms

1. Introductory remarks

Efficient modeling techniques should allow for a selection of pertinent variables and a formation of highly representative datasets. The models should be able to take advantage of the existing domain knowledge (such as a prior experience of human observers or operators) and augment it by available numeric data to form a coherent data-knowledge modeling entity. The omnipresent modeling tendency is the one that exploits techniques of Computational Intelligence (CI) by embracing fuzzy modeling [1-6], neurocomputing [7], and genetic optimization [8,9].

In this study, we develop a hybrid modeling architecture, called genetically optimized Multi-layer Fuzzy Neural Networks (gMFNN). In a nutshell, gMFNN is composed of two main substructures driven to genetic optimization, namely a fuzzy set-based fuzzy neural network (FNN) and a polynomial neural network (PNN). From a standpoint of rule-based architectures, one can regard the FNN as an implementation of the antecedent part of the rules while the consequent (conclusion part) is realized with the aid of a PNN. The role of the FNN is to interact with input data, granulate the corresponding input spaces (viz. converting the numeric data into representations at the level of fuzzy sets). In the first case (Scheme I) we concentrate on the use of simplified fuzzy inference. In the second case (Scheme II), we take advantage of linear fuzzy inference. The role of the PNN is to carry out nonlinear transformation at the level of the fuzzy sets formed at the level of FNN. The PNN that exhibits a flexible and versatile structure [10] is constructed on a basis of a Group Method of Data Handling (GMDH [21]) method and genetic algorithms (GAs). In this network, the number of layers and number of nodes in each layer are not predetermined but can be generated in a dynamic fashion. The design procedure applied in the construction of each layer of the PNN deals with its structural optimization involving the selection of optimal nodes with specific local characteristics (such as the number of input variables, the order of the polynomial, and a collection of the specific subset of input variables) and addresses specific aspects of parametric optimization. To assess the performance of the proposed model, we exploit a well-known time series data. Furthermore, the network is directly contrasted with several existing intelligent models.

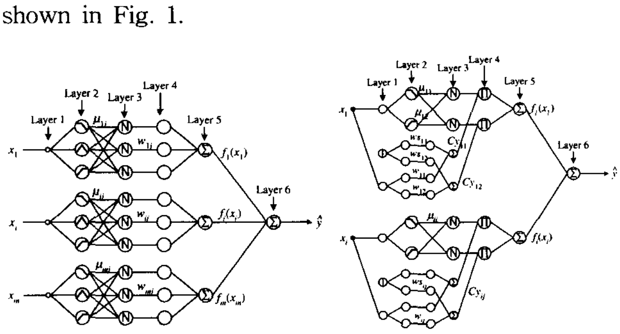
접수일자 : 2004년 3월 31일
완료일자 : 2004년 5월 28일
감사의 글 : This work has been supported by KESRI (R-2003-0-285), which is funded by MOCIE (Ministry of commerce, industry and energy).

2. The architecture and development of genetically optimized MFNN (gMFNN)

The gMFNN emerges from the genetically optimized multi-layer perceptron architecture based on fuzzy set-based FNN, GAs and the GMDH method. In the sequel, these networks result as a synergy between two other general constructs such as FNN [14,15] and PNN [10].

2.1 Fuzzy neural networks and genetic optimization

We use FNN based on two types of fuzzy inferences, that is, simplified fuzzy inference-based FNN (Scheme I) and linear fuzzy inference-based FNN (Scheme II) as shown in Fig. 1.



(a) Scheme I (b) Scheme II

Fig. 1. Topologies of fuzzy set-based FNN

The notation used in Fig.1 requires some clarification. The "circles" denote units of the FNN while "N" identifies a normalization procedure applied to the membership grades of the input variable x_i. The output of the "Σ" neuron is described by a nonlinear function $f_i(x_i)$. Not necessarily f_i is a sigmoid function encountered conventional neural networks but we allow for more flexibility in this regard. Finally, the output of the FNN is governed by the following expression.

$$\hat{y}=f_1(x_1)+f_2(x_2)+\cdots+f_m(x_m)=\sum_{i=1}^{m} f_i(x_i) \tag{1}$$

with m being the number of the input variables. We can regard each $f_i(x_i)$ given by (1) as the following mappings (rules).

$$\text{Scheme I} \quad R^j: \textit{If } x_i \textit{ is } A_{ij} \textit{ then } Cy_{ij}=w_{ij} \tag{2}$$

$$\text{Scheme II} \quad R^j: \textit{If } x_i \textit{ is } A_{ij} \textit{ then } Cy_{ij}=ws_{ij}+w_{ij}\cdot x \tag{3}$$

R^j is the $j-th$ fuzzy rule while A_{ij} denotes a fuzzy variable of the premise of the fuzzy rule and represents a membership function μ_{ij}. wij is a constant in (2), and ws_{ij} is a constant and wij is an input variable consequence of the fuzzy rule in (3). They express a connection (weight) existing between the neurons as visualized in Fig. 1. Mapping from x_i to $f_i(x_i)$ in (2) is determined by the fuzzy inferences and a standard defuzzification (center of gravity aggregation).

$$f_i(x_i)=\sum_{j=1}^{z}\mu_{ij}(x_i)\cdot w_{ij} \Big/ \sum_{j=1}^{z}\mu_{ij}(x_i) \tag{4}$$

The learning of FNN is realized by adjusting connections of the neurons and as such it follows a standard Back-Propagation (BP) algorithm [14]. For the simplified fuzzy inference-based FNN, the update formula of a connection in Scheme I is as follow.

$$\begin{aligned}\Delta w_{ij} = \; & 2\cdot\eta\cdot(y_p-\hat{y}_p)\cdot\mu_{ij}(x_i)\\ & +\alpha(w_{ij}(t)-w_{ij}(t-1))\end{aligned} \tag{5}$$

where, y_p is the p-th target output data, $\hat{y}_p$ stands for the $p-th$ actual output of the model for this specific data point, η is a positive learning rate and α is a momentum coefficient constrained to the unit interval. The inference result and the learning algorithm in linear fuzzy inference-based FNN use the mechanisms in the same manner as discussed above.

The task of optimizing any complex model involves two main phases. First, a class of some optimization algorithms has to be chosen so that it is applicable to the requirements implied by the problem at hand. Secondly, various parameters of the optimization algorithm need to be tuned in order to achieve its best performance. Along this line, genetic algorithms (GAs) are optimization techniques based on the principles of natural evolution. In essence, they are search algorithms that use operations found in natural genetics to guide a comprehensive search over the parameter space [8,9]. In order to enhance the learning of the FNN and augment its performance of a FNN, we use GAs to adjust learning rate, momentum coefficient and the parameters of the membership functions of the antecedents of the rules.

2.2 Genetically optimized PNN (gPNN)

When we construct PNs of each layer in the conventional PNN [10], such parameters as the number of input variables (nodes), the order of polynomial, and input variables available within a PN are fixed (selected) in advance by the designer. This could have frequently contributed to the difficulties in the design of the optimal network.

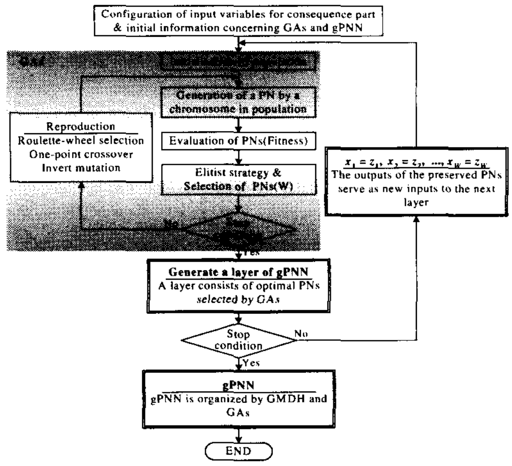


Fig. 2. Overall genetically-driven optimization process of PNN

To overcome this apparent drawback, we introduce a new genetic design approach; especially as a consequence we will be referring to these networks as genetically optimized PNN (to be called gPNN). The overall genetically-driven optimization process of PNN is shown in Fig. 2.

3. The algorithms and design procedure of genetically optimized MFNN

The premise of gMFNN: FNN (Refer to Fig. 1)

[Layer 1] Input layer.

[Layer 2] Computing activation degrees of linguistic labels.

[Layer 3] Normalization of a degree activation of the rule.

[Layer 4] Multiplying a normalized activation degree of the rule by connection (weight).

$$a_{ij} = \overline{\mu_{ij}} \times Cy_{ij} = \mu_{ij} \times Cy_{ij} \tag{6}$$

Simplified : $Cy_{ij} = w_{ij}$

Linear : $Cy_{ij} = ws_{ij} + w_{ij} + w_{ij} \cdot x_i$ (7)

If we choose Connection point 1 for combining FNN with gPNN as shown in Fig. 3, aij is given as the input variable of the gPNN.

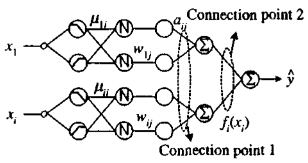


Fig. 3. Connection points for combining FNN with gPNN

[Layer 5] Fuzzy inference for the fuzzy rules. If we choose Connection point 2, fi is the input variable of gPNN.

[Layer 6; Output layer of FNN] Computing output of a FNN.

The consequence of gMFNN: gPNN (Refer to Fig. 2)

[Step 1] Configuration of input variables. If we choose the first option, $x_1 = a_{11}$, $x_2 = a_{12}$, $\cdots$, $x_n = a_{ij}$ $(n = i \times j)$. For the second option, we have $x_1 = f_1$, $x_2 = f_2$, $\cdots$, $x_n = f_m$ $(n = m)$.

[Step 2] Decision of initial information for constructing the gPNN.

[Step 3] Initialization of population.

[Step 4] Decision of PNs structure using genetic design.

This concerns the selection of the number of input variables, the polynomial order, and the input variables to be assigned in each node of the corresponding layer. We divide the chromosome to be used for genetic optimization into three sub-chromosomes as shown in Fig. 4. The 1^{st} sub-chromosome contains the number of input variables, the 2^{nd} involves the order of the polynomial of the node, and the 3^{rd} (remaining bits) contains input variables coming to the corresponding node (PN). In Fig. 5, 'PNn' denotes the n^{th} PN (node) of the corresponding layer, 'N' denotes the number of inputs coming to the node, and 'T' denotes the polynomial order in the node[10].

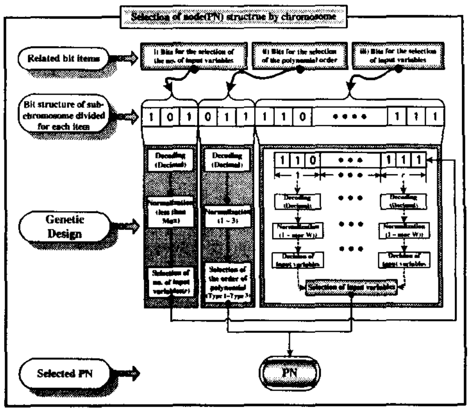


Fig. 4. The PN design using genetic optimization

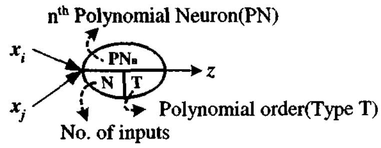


Fig. 5. Formation of each PN

[Step 5] Evaluation of PNs.

[Step 6] Elitist strategy and selection of PNs with the best predictive capability. We select W of PNs characterized by the best fitness values.

[Step 7] Reproduction. To move on to the next generation, we carry out selection, crossover, and mutation operation using genetic information and the fitness values obtained via Step 5.

[Step 8] Repeating Step 4-7.

[Step 9] Construction of their corresponding layer.

[Step 10] Check the termination criterion. In this study, we use a measure (performance index) that is the Mean Squared Error (MSE).

$$E(PI \ or \ EPI) = \frac{1}{n} \sum_{p=1}^{n} (y_p - \hat{y}_p)^2 \tag{8}$$

[Step 11] Determining new input variables for the next layer.

The gPNN algorithm is carried out by repeating Steps 4-11.

4. Experimental studies

The performance of the gMFNN is illustrated with the aid of a time series of gas furnace (Box-Jenkins data [16]). Genetic algorithms use binary type, roulette-wheel selection, one-point crossover, and an invert operation in the mutation. The crossover rate of GAs is set to 0.75 and probability of mutation is equal to 0.065. The time series data (296 input-output pairs) resulting from the gas furnace process has been intensively studied in the previous literature [1-6,12-16]. The delayed terms of methane gas flow rate, $u(t)$ and carbon dioxide density,

$y(t)$ are used as system input variables. We use two types of system input variables of FNN structure, Type I and Type II to design an optimal model from gas furnace data. Type I utilize two system input variables such as $u(t-3)$ and $y(t-1)$ and Type II utilizes 3 system input variables such as $u(t-2)$, $y(t-2)$, and $y(t-1)$. The output variable is $y(t)$. The first part of the dataset (consisting of 148 pairs) was used for training. The remaining part of the series serves as a testing set.

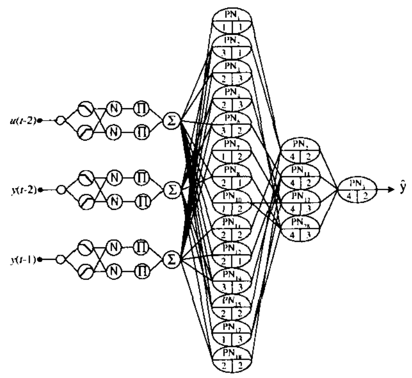
Table 1. Parameters of the optimization environment and computational effort

GAs	Generation			150
	Population size			60
	Elite population size (W)			30
	String length	Premise structure (FNN)		10 (per one variable)
		Consequence structure (PNN)		3+3+24
gMFNN	Premise (FNN)	No. of entire system inputs		3
		Learning iteration		500
		Learning rate tuned	Simplified	0.0124
			Linear	0.0204
		Momentum Coefficient tuned	Simplified	0.0083
			Linear	0.0094
		No. of rules		6
	Consequence (gPNN)	No. of entire inputs	Connection point 1	6
			Connection point 2	3
		Maximal layer		5
		No. of inputs to be selected(N)		$1 \leq N \leq 4$
		Type(T)		$1 \leq T \leq 3$

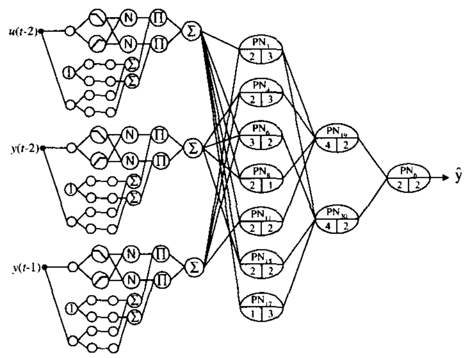
N, T : integer

Table 2. Performance index of gMFNN for the gas furnace

Premise part				CP	Consequence part								
FIS	No. of rules (MFs)	PI	EPI		Layer	No. of inputs	Input No.				T	PI	EPI
Simplified	6 (2+2+2)	0.0248	0.126	01	1	4	5	2	1	3	3	0.0220	0.135
					2	4	15	28	26	3	2	0.0209	0.135
					3	4	5	1	25	16	3	0.0205	0.131
					4	4	1	2	28	26	2	0.0190	0.128
					5	4	18	6	28	1	2	0.0175	0.125
				02	1	3	2	3	2	·	3	0.0221	0.135
					2	4	11	2	15	13	2	0.0194	0.126
					3	4	28	2	11	17	2	0.0190	0.116
					4	4	2	6	23	27	1	0.0188	0.114
					5	4	1	12	7	26	3	0.0182	0.112
Linear	6 (2+2+2)	0.0256	0.143	01	1	4	6	3	1	5	3	0.0218	0.136
					2	4	6	24	16	30	3	0.0197	0.124
					3	3	4	16	26	·	1	0.0196	0.121
					4	4	22	24	1	13	1	0.0193	0.119
					5	3	11	18	21	·	3	0.0191	0.117
				02	1	3	1	2	3	·	3	0.0232	0.130
					2	4	12	15	13	6	2	0.0196	0.120
					3	2	19	30	·	·	2	0.0194	0.115
					4	4	2	21	11	5	1	0.0188	0.113
					5	4	13	3	26	25	1	0.0184	0.110



(a) Simplified fuzzy inference



(b) Linear fuzzy inference

Fig. 6. Optimal topology of gMFNN for the gas furnace

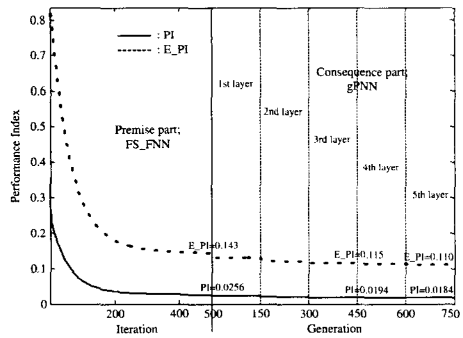


Fig. 7. Optimization procedure of gMFNN by BP learning and GAs

Table 1 shows computational aspects related to the genetic optimization of the network. Table 2 summarizes the results of the optimized architectures according to connection points based on each fuzzy inference method. In Table 2, as the premise FNN, Type II is shown. The values of the performance index of output of the gMFNN depend on each connection point based on the individual fuzzy inference methods. The values of the performance index vis-à-vis choice of number of layers of gMFNN related to the optimized architectures in each layer of the network are shown in Table 2. The optimal topology of gMFNN is shown in Fig. 6. Fig. 7 illustrates the optimization process by visualizing the performance index in successive cycles (iteration and generation). It also shows the optimized network architecture when taking into consideration gMFNN based on linear fuzzy inference and connection point (CP) 2, refer to Table 2. Table 3 contrasts the performance of the genetically developed network with other fuzzy and fuzzy-neural networks studied in the literatures. It becomes obvious that the proposed genetically optimized gMFNN architectures outperform other models both in terms of their accuracy and higher generalization capabilities.

Table 3. Comparison of performance with other modeling methods

Model			PI	EPI	No. of rules
Box and Jenkins model [16]			0.710		
Pedryczs model [1]			0.320		
Xu and Zailus model [2]			0.328		
Sugeno and Yasukawa's model [3]			0190		
Kim, et al.'s model [17]			0.034	0.244	2
Lin and Cunningham's mode [18]			0.071	0.261	4
Fuzzy	Complex [4]	Simplified	0.024	0.328	4(2×2)
		Linear	0.023	0.306	4(2×2)
	Hybrid [6] (GAs+ Complex)	Simplified	0.024	0.329	4(2×2)
		Linear	0.017	0.289	4(2×2)
	HCM+GAs [5]	Simplified	0.035	0.289	4(2×2)
			0.022	0.333	6(3×2)
		Linear	0.026	0.272	4(2×2)
			0.020	0.264	6(3×2)
FNN [15]		Simplified	0.043	0.264	6(3+3)
		Linear	0.037	0.273	6(3+3)
SOFPNN		Generic [12]	0.017	0.250	4 rules/ 5th layer
		Advanced [13]	0.019	0.264	6 rules/ 5th layer
Proposed model (gMFNN)		Simplified	0.018	0.254	4 rules/ 5th layer
			0.018	0.112	6 rules/ 5th layer
		Linear	0.015	0.258	4 rules/ 5th layer
			0.016	0.110	6 rules/ 5th layer

6. Concluding remarks

The comprehensive design methodology comes with the parametrically as well as structurally optimized network architecture. 1) As the premise structure of the gMFNN, the optimization of the rule-based FNN hinges on genetic algorithms and back-propagation (BP) learning algorithm: The GAs leads to the auto-tuning of vertexes of membership function, while the BP algorithm helps obtain optimal parameters of the consequent polynomial of fuzzy rules through learning. And 2) the gPNN that is the consequent structure of the gMFNN is based on the technologies of the extended GMDH algorithm and GAs: The extended GMDH method is comprised of both a structural phase such as a self-organizing and evolutionary algorithm (rooted in natural law of survival of the fittest), and a parametric phase of least square estimation (LSE)-based learning, moreover the gPNN architecture is driven to genetic optimization, in what follows it leads to the selection of the optimal nodes (or PNs) with local characteristics such as the number of input variables, the order of the polynomial, and a collection of the specific subset of input variables. In the sequel, a variety of architectures of the proposed gMFNN driven to genetic optimization have been discussed. The experiments helped compare the network with other intelligent models - in all cases the previous models came with higher values of the performance index.

References

[1] W. Pedrycz, "An identification algorithm in fuzzy relational system," Fuzzy Sets and Systems, Vol. 13, pp.153-167, 1984.

[2] C. W. Xu, and Y. Zailu, "Fuzzy model identification self-learning for dynamic system," IEEE Trans. on Syst. Man, Cybern., Vol. SMC-17, No.4, pp.683-689, 1987.

[3] M. Sugeno and T. Yasukawa, "A Fuzzy-Logic-Based Approach to Qualitative Modeling," IEEE Trans. Fuzzy Systems, Vol. 1, No. 1, pp. 7-31, 1993.

[4] S.-K. Oh and W. Pedrycz, "Fuzzy Identification by Means of Auto-Tuning Algorithm and Its Application to Nonlinear Systems," Fuzzy Sets and Systems, Vol. 115, No. 2, pp. 205-230, 2000.

[5] 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, Issue 5, pp. 406-418, 2001.

[6] S.-K. Oh, W. Pedrycz and B.-J. Park, "Hybrid Identification of Fuzzy Rule-Based Models," International Journal of Intelligent Systems, Vol. 17, Issue 1, pp. 77-103, 2002.

[7] K. S. Narendra and K. Parthasarathy, "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks," IEEE Transactions on Neural Networks, Vol. 2, pp. 252-262, 1991.

[8] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs," Springer-Verlag, Berlin Heidelberg, 1996.

[9] Holland, J. H., "Adaptation in Natural and Artificial Systems," The University of Michigan Press, Ann Arbour, 1975.

[10] S.-K. Oh, W. Pedrycz and B.-J. Park, "Polynomial Neural Networks Architecture: Analysis and Design," Computers and Electrical Engineering,

Vol. 29, Issue 6, pp. 653-725, 2003.

[11] A. G. Ivahnenko, "The group method of data handling: a rival of method of stochastic approximation," Soviet Automatic Control, Vol. 13, No. 3, pp. 43-55, 1968.

[12] B.-J. Park, S.-K. Oh, S.-W. Jang, "The Design of Adaptive Fuzzy Polynomial Neural Networks Architectures Based on Fuzzy Neural Networks and Self-Organizing Networks," Journal of Control, Automation and Systems Engineering, Vol. 8, No. 2, pp.126-135, 2002. (In Korean)

[13] B.-J. Park and S.-K. Oh, "The Analysis and Design of Advanced Neurofuzzy Polynomial Networks," Journal of the Institute of Electronics Engineers of Korea, Vol. 39-CI, No. 3, pp.18-31, 2002. (In Korean)

[14] T. Yamakawa, "A New Effective Learning Algorithm for a Neo Fuzzy Neuron Model," 5th IFSA World Conference, pp. 1017-1020, 1993.

[15] S.-K. Oh, W. Pedrycz and H.-S. Park, "Hybrid Identification in Fuzzy-Neural Networks," Fuzzy Sets and Systems, Vol. 138, No. 2, pp. 399-426, 2003.

[16] D. E. P. Box and G. M. Jenkins, "Time Series Analysis, Forecasting, and Control," 2nd edition Holden-Day, SanFransisco, 1976.

[17] E. Kim, H. Lee, M. Park and M. Park, "A Simply Identified Sugeno-type Fuzzy Model via Double Clustering," Information Sciences, Vol. 110, pp. 25-39. 1998.

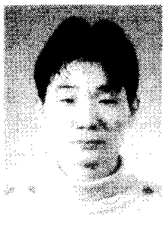
[18] Y. Lin, G. A. Cunningham III, "A new Approach to Fuzzy-neural Modeling," IEEE Transaction on Fuzzy Systems, Vol. 3, No. 2, pp. 190-197, 1997.

[19] S.-K. Oh, Computational Intelligence by Programming(focused on Fuzzy, Neural Networks, and Genetic Algorithms), Naeha press, 2002.

저 자 소 개

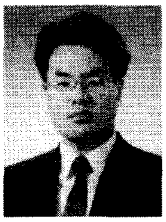
Byoung-Jun Park

He received the B.S., and M.S. degrees in control and instrumentation engineering from the Wonkwang University, in 1998, and 2000, respectively. He received his Ph.D. degrees from the Department of Electrical Engineering, Wonkwang University, in 2003. His research interests concerns fuzzy, neurofuzzy systems, genetic algorithms, Computational Intelligence, hybrid systems, and intelligent control.



Keon-Jun Park

He received the B.S degrees in electronics engineering from the Wonkwang University, in 2003. He is now pursuing his Master degree in control and instrumentation engineering, Wonkwang University. His research interests include FIS, Neural Networks, GAs, optimization theory, Computational Intelligence and Automation control.



Dong-Yoon Lee

He received the B.S. degree in the Department of Electrical Engineering from Wonkwang University, in 1987. And he received the M.S., Ph.D. degrees in the Department of Electrical Engineering from Yonsei University, in 1990, 2001, respectively. He had worked as a research engineer in Intellect, J-tek and KIST from 1990 to 2000. From March 2001 to February 2002, he was a B.K. Professor in the School of Electrical, Electronic and Information Engineering, Wonkwang University. He is currently a assistant professor in the Department of Information and Communication Engineering, Joongbu University.

Sung-Kwun Oh

He received the B.S., M.S., and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, Korea, in 1981, 1983, and 1993, respectively. During 1983-1989, he had worked as the Senior Researcher of R&D Lab. of Lucky-Goldstar Industrial Systems Co., Ltd. He had worked as a Postdoctoral fellow in the Department of Electrical and Computer Engineering University of Manitoba, Canada, from 1996 to 1997. He is currently an Associate Professor in the School of Electrical, Electronic and Information Engineering, Wonkwang University, Korea. His research interests include the fields of fuzzy system, fuzzy-neural networks, automation systems, advanced Computational Intelligence, and intelligent control. He is a member of IEEE. He currently serves as a Associate Editor of the Korean Institute of Electrical Engineers(KIEE) and the Institute of Control, Automation & Systems Engineers(ICASE), Korea