Fuzzy Logic-Based Networks: A Study in Logic Data Interpretation

Xiaofeng Liang,[1,*] Witold Pedrycz[1,2,†]
[1]*Department of Electrical & Computer Engineering, University of Alberta, Edmonton T6R 2G7, Canada*
[2]*Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland*

Fuzzy neurons may have outstanding learning abilities and are endowed with significant interpretation capabilities. In this study, we are concerned with the development of logic networks composed of fuzzy neurons. The main phase of the design includes the granulation of the output space (via triangular fuzzy sets) being realized with the use of fuzzy equalization. In the sequel these fuzzy sets are used to guide the construction of a family of fuzzy sets in the input space. Further processing of the resulting fuzzy sets deals with some additional aggregation of those that are not sufficiently distinct. This helps reduce the size of the logic network. We include comprehensive experimentation and offer a thorough interpretation of the networks. Experiments concerning real-world continuous data help evaluate the network's appealing properties: transparent interpretability and practical feasibility. © 2006 Wiley Periodicals, Inc.

1. INTRODUCTION

Fuzzy modeling is aimed at constructing interpretable systems that typically result in a collection of "if–then" rules. Compared with such rule-based modeling, neural networks possess superb learning abilities, including a vast variety of supervised and unsupervised learning schemes. From the early studies on neurofuzzy systems (cf. Refs. 1 and 2), we came to a clear realization of the importance of the synergy between fuzzy sets and neural networks. We have witnessed a great deal of effort being spent on the design of neurofuzzy systems and networks; refer, for example, to Refs. 3–8. The ultimate trade-off between accuracy and interpretability (transparency) is of vital interest when dealing with the conceptualization, design, learning, and validation of neurofuzzy systems.

Fuzzy (logic) neurons[9,10] are simple processing units that emphasize the synergy of transparency and learning abilities to a very high extent.[11] Given the ensuing transparency of the generic constructs (neurons), further architectural developments are quite intuitive. For instance, fuzzy logic neural networks

*Author to whom all correspondence should be addressed: e-mail: xfliang@ece.ualberta.ca.
†e-mail: pedrycz@ece.ualberta.ca

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 21, 1249–1267 (2006)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.20186

(FLNN)[12–14] encompass a number of neurons, of both AND and OR types that, when organized into a series of layers, generalize the well-known Shannon representation theorem of Boolean functions. Likewise, so-called OR/AND neurons help address the needs for higher flexibility of individual neurons. Such a need is triggered by the complexity of semantics of logic connectives in fuzzy sets that goes beyond the level of plain *and* and *or* operators.

A construction of interpretable and meaningful linguistic labels (fuzzy sets) is a critical development phase. In the quest for such constructs, we have to become fully cognizant that fuzzy sets should be both (a) conceptually meaningful (semantically sound) and (b) experimentally justifiable. To address these two important requirements, we elaborate on the matter of fuzzy equalization.[15] The processing core of fuzzy modeling that revolves around constructs of logic neurons deserves our attention, as they lead directly to the explicit interpretation of numeric data.

In a nutshell, there are two key objectives of this study:

- To develop an extensive design platform for logic-based models that includes numeric interfaces used to construct a collection of meaningful and experimentally sound information granules (fuzzy relations and fuzzy sets).
- To complete a comprehensive suite of numeric experiments to arrive at a logical description of numeric data.

The organization of the article reflects the underlying research agenda. First, in Section 2, we present a concise description of logic-driven fuzzy neurons, together with a topology of networks formed by these logic neurons. Section 3 discusses the design issues, which concern the granulation of data with the use of fuzzy equalization and specialized mechanisms of fuzzy clustering, a so-called conditional fuzzy C-Means, to be described in more detail. In the sequel, we elaborate on possible reduction of the number of fuzzy sets coming from the projection of the results of the fuzzy clustering. We also cover the mechanisms of gradient-based learning. The interpretation of the networks is presented in Section 4. Section 5 deals with experimentation using real-world data and focuses on the interpretation of the resulting models (logic expressions). Conclusions are given in Section 6.

2. FUZZY NEURONS AND FUZZY LOGIC-BASED NEURAL NETWORKS

In this section, we briefly discuss the basic processing units—logic neurons[9,10]—and then show how they give rise to some architectures of a clearly emphasized logic character.

2.1. AND and OR Fuzzy Neurons

AND and OR neurons are examples of fuzzy neurons that realize some *and*- and *or*-like logic aggregation of inputs. Their realization hinges upon the use of t-norms and t-conorms (s-norms). The AND neuron is generally a nonlinear logic processing element with n inputs $\mathbf{x} \in [0,1]^n$ governed by the following relationship:

$$y = \mathrm{AND}(\mathbf{x}; \mathbf{w}) \tag{1}$$

where $\mathbf{w}$ stands for an n-dimensional vector of adjustable connections (or weights) whose values are in the unit interval. The composition of $\mathbf{x}$ and $\mathbf{w}$ is realized by a certain t-s composition that makes use of some t- and s-conorms. The aggregation consists of two steps. In the first step inputs are combined with the corresponding weights. In the sequel we aggregate these partial results with the aid of some t-norm. In other words we obtain

$$y = \mathrm{AND}(\mathbf{x}; \mathbf{w}) = \mathop{T}_{i=1}^{n} [w_i s x_i] \tag{2}$$

Here s and t stand for the t-conorms and the t-norms, respectively. We can rewrite (2) in an equivalent format that facilitates its further interpretation:

$$y \text{ is } (x_1)_{w_1} \text{ and } (x_2)_{w_2} \ldots \text{ and } (x_n)_{w_n} \tag{3}$$

where $(x_i)_{w_i}$ is a shorthand notation for $(x_i \; or \; w_i)$, $i = 1, \ldots, n$.

Note that the lower the value of the connection, the more essential the corresponding input. In the limit, we view the input meaningless if the associated connection is equal to 1.

By reversing the order of the t-norms and t-conorms in Equation 2, we arrive at the construct of an OR neuron:

$$y = \mathrm{OR}(\mathbf{x}; \mathbf{w}) = \mathop{S}_{i=1}^{n} [w_i t x_i] \tag{4}$$

For interpretation purposes, we use the notation

$$y \text{ is } [x_1]_{w_1} \text{ or } [x_2]_{w_2} \ldots \text{ or } [x_n]_{w_n} \tag{5}$$

where $[x_i]_{w_i}$ denotes $(x_i \; and \; w_i)$, $i = 1, \ldots, n$.

The higher the value of the connection, the more relevant the corresponding input. In the limit, if $w_i = 0$, the ith input x_i can be fully eliminated.

2.2. The Topology of Fuzzy Logic-Based Neural Networks

By arranging AND and OR neurons in layers, we arrive at the generic logic realization of some logic input–output mapping. The so-called sum of minterms (SOM) network that generalizes the well-known representation scheme being used in the realization of Boolean functions is shown in Figure 1.

This network is uniquely characterized by such parameters as the number of inputs (n), number of AND nodes located in the hidden layer comprising (h) processing units. The connections are organized in some matrix $\mathbf{V}$ and the vector of the connections of the OR neuron is denoted here by $\mathbf{w}$. The network's formulas is written down in the form

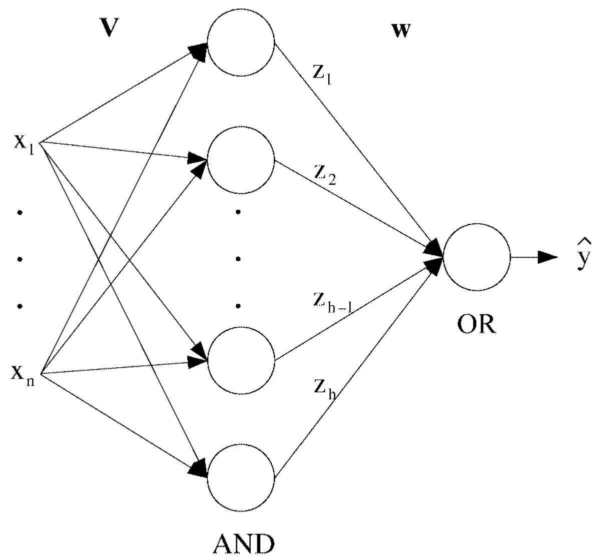

Figure 1. A topology of the SOM type of the logic network; shown are also the connections of the neurons.

$$z_j = \mathrm{AND}(\mathbf{x}, \mathbf{v}_j), \quad j = 1, 2, \ldots, h \tag{6}$$
$$\hat{y} = \mathrm{OR}(\mathbf{z}, \mathbf{w})$$

where $\mathbf{z}$ is a vector of outputs of the AND neurons ($\mathbf{z} = (z_1, z_2, \ldots, z_h)^T$) and $\mathbf{v}_j$ denotes the jth column of the connection matrix $\mathbf{V}$.

As mentioned earlier, the connections of the logic neurons come with a clear interpretation. Owing to the monotonicity property of t-norms and t-conorms, we can proceed with pruning of the weakest connections.

For AND neurons, the weakest connections (which are those above some threshold μ) are converted to 1:

$$v_\mu = \begin{cases} v & v \le \mu \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

For the OR neuron, we use the following relationship to set the values of the weakest connections to zero:

$$w_\lambda = \begin{cases} w & w \ge \lambda \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\mu, \lambda \in [0,1]$; v, w denote some connections of the neurons. By changing the values of the thresholds μ and λ, we alter the "intensity" of pruning of the overall network.

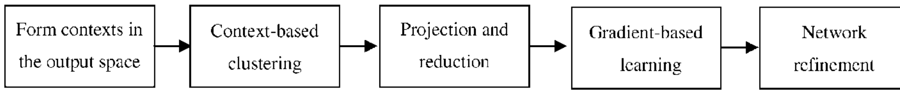

Figure 2. An overall design architecture.

3. THE DESIGN OF THE NETWORK

In this section, we will focus on the key design phases of the network, namely (1) forming contexts in the output space through the mechanism of fuzzy equalization, (2) context-based clustering applied to the input variables, (3) projection and reduction for each variable, (4) gradient-based learning of the connections of the network, and finally (5) refinement of the network through some interpretability mechanisms. Figure 2 shows the overall design architecture.

3.1. Formation of Contexts through Fuzzy Equalization

The underlying idea is to construct fuzzy sets in such a way that they come with a clearly defined semantics and are experimentally justifiable. Fuzzy equalization[15] helps construct linguistic labels (fuzzy sets) that are both semantically and experimentally legitimate. With the fuzzy equalization completed in the output space, we end up with p contexts (fuzzy sets). We assume that these fuzzy sets are described by triangular membership functions with an overlap of 0.5 between two successive linguistic terms. Furthermore denote the family of fuzzy sets as $\mathbf{A} = \{A_1, A_2, \ldots, A_p\}$. Each of these fuzzy sets is fully described by its three parameters describing its modal value (denoted by m) and the lower and upper bound, respectively. We denote these by a and b.

Assume that the probability density function (pdf) of the output is given by $p(y)$. Start from the lower bound of y denoted by $y_{\min}$. Then the parameters of each fuzzy set are computed as shown in Table I; for more details refer to Ref 15.

Note that for the discrete data set $\mathbf{Y} = \{y_1, y_2, \ldots, y_N\}$, the calculations of the probability of A, $P(A)$ is computed through the summation of the discrete probability values.

3.2. Conditional Fuzzy C-Means in the Formation of the Blueprint of the Logic Network

Conditional (context-based) Fuzzy C-Means were introduced in Refs. 16 and 17 as a certain modification of the generic Fuzzy C-Means (FCM),[18] which is guided by an auxiliary (conditional) variable. The method reveals a structure within a family of data by considering their vicinity in a feature space along with the similarity of the associated values assumed by a certain conditional variable. The algorithmic underpinnings go as follows. Assume that $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ are n-dimensional data defined in $\mathbf{R}^n$, and we have been provided with p contexts (say, context-1, ..., context p), being the result of the fuzzy equalization described in the previous section. $\{f_{kj}\}$ are the membership values for each context when

Table I. Fuzzy equalization for triangular fuzzy sets $A_1, A_2, \ldots, A_p$.

$A_1(a,m,b)$	a	$y_{\min}$	
	m	$\int_{y_{\min}}^{a} A_1(y)p(y)\,dy = \int_{y_{\min}}^{a} p(y)\,dy$	
	b	$\int_{m}^{b} A_1(y)p(y)\,dy = \frac{1}{2p}$	
$A_2(a,m,b)$	a	m of A_1	
	m	b of A_1	
	b	$\varepsilon = \int_{m}^{b} A_1(y)p(y)\,dy$	$\int_{m}^{b} A_2(y)p(y)\,dy = \frac{1}{p} - \varepsilon$
...	...		...
$A_p(a,m,b)$	a	m of A_{p-1}	
	m	b of A_{p-1}	
	b	$y_{\max}$	

Shown are formulas for the parameters of the corresponding fuzzy sets.

we apply the output y through these p fuzzy sets. Table II summarizes the complete algorithm.

3.3. Projection and Reduction for Input Variables

For each context in the output space, we have generated c corresponding prototypes (clusters) in the input space. Thus for p contexts we end up with $c * p$ prototypes (clusters) as schematically displayed in Figure 3.

Each prototype is then projected onto the individual variables of the input space. Along with the minimum and maximum values of each variable, the coordinates of the prototypes in the corresponding input space form $c * p + 2$ fuzzy sets. As an example, for two contexts with two clusters per context, a relationship between the fuzzy sets of contexts and the resulting fuzzy sets arising for each input is visualized in Figure 4.

We merge fuzzy sets if their modal values are close to each other (which make these fuzzy sets quite indistinguishable). The merging is guided by the following aggregation criterion.

Consider a certain input variable, say x; its lower and upper bound are denoted as min and max, respectively. The coordinates of $c * p$ prototypes result in $c * p + 2$ fuzzy sets $\{A_1, A_2, \ldots, A_{c*p+2}\}$ built in the region of [min, max]. Define D as a threshold measure ($D = (\max - \min)/(c \cdot p)$); evaluate the distance between any two successive fuzzy sets. We merge two successive fuzzy sets by forming a single fuzzy set with a trapezoidal membership function if the distance satisfies the following merging criterion:

$$\|A_{i+1} - A_i\| < \varepsilon \cdot D, \quad i = 1, 2, \ldots, c \cdot p + 1 \tag{9}$$

where $\varepsilon \in [0,1]$ and $\|\cdot\|$ is the Euclidean distance of the centers of the two successive fuzzy sets.

Table II. Conditional Fuzzy C-Means: a flow of computing.

Given:	Dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathbf{R}^n$ Membership values for context-j: $f_{kj}(k = 1, 2, \ldots, N, j = 1, \ldots, p)$
Specified:	The number of prototypes c $(1 < c < N)$, exponential weight $m(1 < m < \infty)$, the termination criterion $\varepsilon(\varepsilon > 0)$, and the distance function $\|\cdot\|$
Initialization	Randomly initialize partition matrix U: $U_j^{(0)} = [u_{ik}]_j$ $(i = 1, \ldots, c, k = 1, \ldots, N)$
Processing	For $j = 1, 2, \ldots, p$ Iterate $iter = 1, 2, \ldots$ *and compute* Prototypes $\mathbf{v}_{ij}$: $v_{ij} = \dfrac{\sum_{k=1}^{N} u_{ik} x_k}{\sum_{k=1}^{N} u_{ik}^m}$ Partition matrix U_j: $u_{ik} = \dfrac{f_{kj}}{\sum_{l=1}^{c} \left(\dfrac{\|x_k - v_{ij}\|}{\|x_k - v_{lj}\|} \right)^{2/(m-1)}}$ Until $\|U_j^{(iter+1)} - U_j^{(iter)}\| < \varepsilon$
Results	Prototypes$_j$ and partition matrix$_j$ $(j = 1, \ldots, p)$

For instance, as a result of such merging, see Figure 4a; instead of two triangular fuzzy sets A_4 and A_5, we consider a single trapezoidal fuzzy set C_1. Given this, we update the relationships replacing the fuzzy sets that have been merged by their new generalized version; see Figure 5 for more details. The structures shown in Figure 5b are then used as a blueprint to form the fuzzy logic network. Each prototype inside the context represents an AND neuron, aggregated by an OR neuron to form the output as the corresponding context.

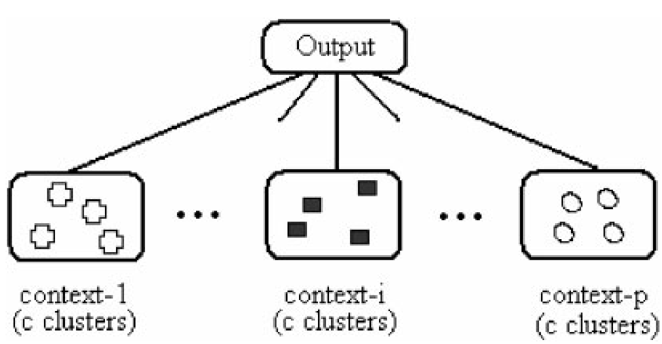

Figure 3. A concept of the context-based clustering; note that each context induces c clusters in the input space.

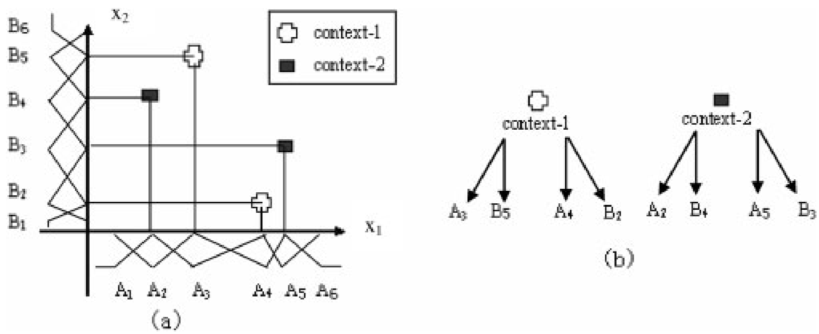

Figure 4. Projection of prototypes along with the contexts and induced clusters in the input space.

Note that, with such projecting and reduction measures, a different number of fuzzy sets may be used for each input variable. This reflects the nature of each physical variable in a most reasonable sense.

3.4. Gradient-Based Learning

The context-based clustering and projection followed by the reduction of the number of fuzzy sets give rise to the structure of the logic network. The optimization of the connections of the network falls under the rubric of parametric optimization and is realized through standard gradient-based learning. This optimization is guided by the standard performance index treated as the mean squared error (MSE),

$$Q = \frac{1}{N}\sum_{k=1}^{N}(y(k) - \hat{y}(k))^2 \tag{10}$$

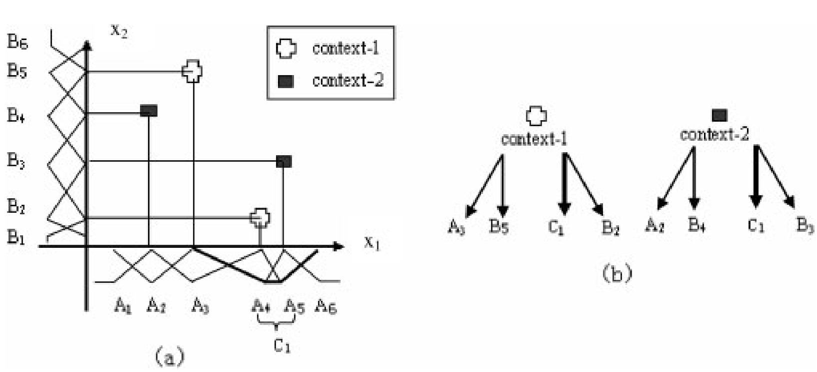

Figure 5. Merging of close fuzzy sets and construct of contexts using updated fuzzy sets.

where N is the number of data patterns, $\hat{y}$ is the output of the network, and y is the required (target) value of the output coming from the data set. Alternatively, the RMS version $E = \sqrt{Q}$ can be used to quantify the performance of the network.

Denoting all connections of the network by **conn**, the updates of their values are governed by the well-known expression

$$\mathbf{conn}(iter + 1) = \mathbf{conn}(iter) - \alpha \nabla_{\mathbf{conn}} Q \tag{11}$$

with α being a positive learning rate. We initialize the weights of the OR neurons to random values close to 1, say those in the interval $[0.9, 1]$. The connections of the AND neurons are initialized to values close to 0, such as those lying within the $[0, 0.1]$ interval. This form of initialization we have emphasizes the relevance of all connections of the network; we anticipate that during learning some connections could be made irrelevant.

If we confine ourselves to the product (t-norm) and probabilistic sum (t-conorm), we can proceed with the detailed computations of the gradient needed for updating the connections. The expressions for values at the neurons can be then written down as follows (cf. Figure 1):

$$\begin{cases} z_j = \prod_{i=1}^{n} (x_i + v_{ij} - x_i v_{ij}), \quad j = 1, \ldots, h \\ \hat{y} = 1 - \prod_{j=1}^{h} (1 - z_j w_j) \end{cases} \tag{12}$$

In the sequel we obtain

$$\mathbf{v}(iter + 1) = \mathbf{v}(iter) - \alpha \nabla_{\mathbf{v}} Q$$

$$\mathbf{w}(iter + 1) = \mathbf{w}(iter) - \alpha \nabla_{\mathbf{w}} Q$$

where

$$\begin{aligned} \nabla_{\mathbf{v}} Q = \frac{\partial Q}{\partial v_{ij}} &= -\frac{1}{2N} \sum_{k=1}^{N} (y_k - \hat{y}_k) \frac{\partial \hat{y}_k}{\partial v_{ij}} \\ &= -\frac{1}{2N} \sum_{k=1}^{N} (y_k - \hat{y}_k) \left[w_j \prod_{\substack{m=1 \\ m \neq j}}^{h} (1 - z_m w_m) \right] \\ &\quad \times \left[\prod_{\substack{m=1 \\ m \neq i}}^{n} (x_m + v_{mj} - x_m v_{mj})(1 - x_i) \right] \\ \nabla_{\mathbf{w}} Q = \frac{\partial Q}{\partial w_j} &= \frac{1}{2N} \sum_{k=1}^{N} (y_k - \hat{y}_k) \frac{\partial \hat{y}_k}{\partial w_j} \\ &= -\frac{1}{2N} \sum_{k=1}^{N} (y_k - \hat{y}_k) \left[z_j \prod_{\substack{m=1 \\ m \neq j}}^{h} (1 - z_m w_m) \right] \end{aligned} \tag{13}$$

4. INTERPRETATION OF THE NETWORK

The logic networks come with clearly defined semantics. The learning endows the neurons with numeric connections whose values may be useful for further reduction of the network, in this way improving its interpretability. Let us recall that higher values of the connections of the OR neurons are more essential, whereas the OR connections with lower values can be dropped. The opposite situation occurs for AND neurons: Here the higher values of the connections could be viewed as meaningless and therefore dropped. The network can be rewritten as a collection of composite "if–then" rules

$$\begin{gathered}\text{if}\\ \text{condition}_1 \text{ and condition}_2 \text{ and} \ldots \text{and} \ldots \text{condition}_L\\ \text{or}\\ \text{condition}_{1'} \text{ and condition}_{2'} \text{ and} \ldots \text{and} \ldots \text{condition}_{L'}\\ \textit{or}\\ \ldots\end{gathered} \tag{14}$$

In these expressions, the subconditions in each rule are arranged starting with the lowest value of the connections of the AND neuron. The rules themselves are organized starting with the highest values of the OR neuron. The pruning can be completed in two different ways:

(1) By applying some thresholding mechanisms. By introducing some threshold values λ and μ for OR and AND neurons, respectively, we eliminate all connections whose values are below λ (OR neurons) and above μ (AND neurons).
(2) By admitting some allowable structural complexity of the logic description. Accepting a maximal number of conditions and rules, we eliminate "weaker" rules and conditions produced by the network.

5. EXPERIMENTAL STUDIES

In this section, we report on a number of experiments carried out for selected Machine Learning data sets (http://www.ics.uci.edu/~mlearn/MLRepository.html). The obtained results are presented in a uniform manner by quantifying the approximation abilities of the corresponding models and showing the details of the resulting logic description of the data. We also point out some trade-offs between the accuracy of the logic models and their interpretability. Throughout the experiments, we used 60% of the data for training and the remaining 40% for testing.

5.1. Boston Housing Data

This data set concerns a description of real estate in the Boston area and its price. Each piece of real estate is characterized by a number of features such as

(1) CRIM: per capita crime rate by town
(2) ZN: proportion of residential land zoned for lots over 25,000 sq. ft.
(3) INDUS: proportion of nonretail business acres per town
(4) NOX: nitric oxides concentration (parts per 10 million)
(5) RM: average number of rooms per dwelling
(6) AGE: proportion of owner-occupied units built prior to 1940
(7) DIS: weighted distances to five Boston employment centers
(8) RAD: index of accessibility to radial highways
(9) TAX: full-value property-tax rate per $10,000
(10) PTRATIO: pupil-teacher ratio by town
(11) B: $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
(12) LSTAT: % lower status of the population

The MEDV, a median value of the home expressed in $1000s, is regarded as the output of our model. Following the overall development scheme introduced in this study, we start with fuzzy equalization in the output space. The meaningful fuzzy sets we define there quantify the values of the house as LOW, MEDIUM, and HIGH. These three terms are semantically sound and offer enough discrimination. The histogram of the output shown in Figure 6 is quite symmetrical with the exception of an elongated tail of higher values of the real estate. We note however that these values occur seldom and could be removed from the construction of the fuzzy sets.

We distribute the fuzzy sets in the space by eliminating all data points that are more than 2σ distant from the mean value of the population of all data; note that we make this requirement stronger than the standard one encountered in statistics that uses a 3σ rule. By completing the fuzzy equalization, we end up with the three linguistic labels ($p = 3$) for the output as shown in Figure 7.

Then context-based clustering is applied, followed by projection onto the individual input variables and possible reduction (merging) of the adjacent fuzzy sets. Consider the case of three clusters ($c = 3$) for each context. Figure 8 illustrates the number of fuzzy sets produced for each input variable with respect to different values of ε used in the merging criterion. Here we only present three variables, AGE, RAD, and PTRATIO. The trends of the rest of the variables are quite similar. Without any merging ($\varepsilon = 0$), we have $c * p + 2 = 11$ fuzzy sets for each variable. For higher values of ε, this number of fuzzy sets starts decreasing whereas

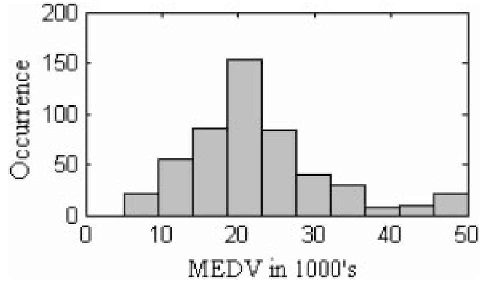

Figure 6. A histogram of MEDV.

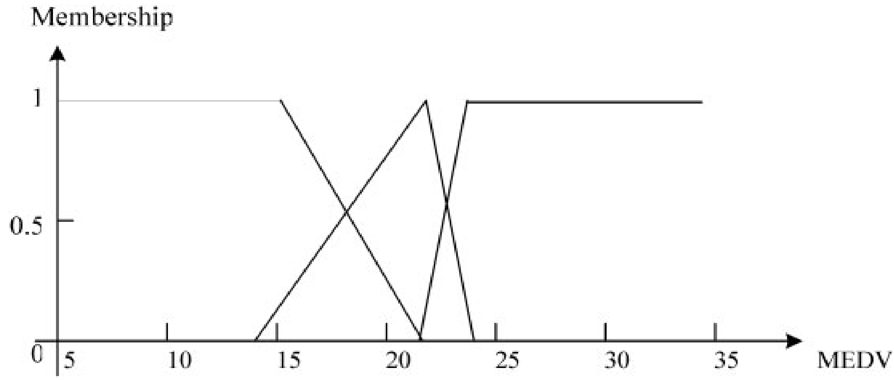

Figure 7. Fuzzy sets constructed in the output space.

the rate of decrease depends on the specific variable. The differences are quite visible, ranging between two and five linguistic terms.

Figure 9 shows the overall number of fuzzy sets (as being counted for all variables) for inputs with three, five, and nine clusters per each context when plotted, versus varying values of ε.

By choosing three clusters for each context along with $\varepsilon = 0.8$ (at which value we obtain a relatively small number of fuzzy sets), see Figure 8, we form the following linguistic terms for each input variable:

(1) CRIM = {LOW, MEDIUM, HIGH} = {L, M, H}
(2) ZN = {LOW, MEDIUM, HIGH, VERY HIGH} = {L, M, H, VH}
(3) INDUS = {VERY LOW, LOW, MEDIUM, HIGH, VERY HIGH} = {VL, L, M, H, VH}

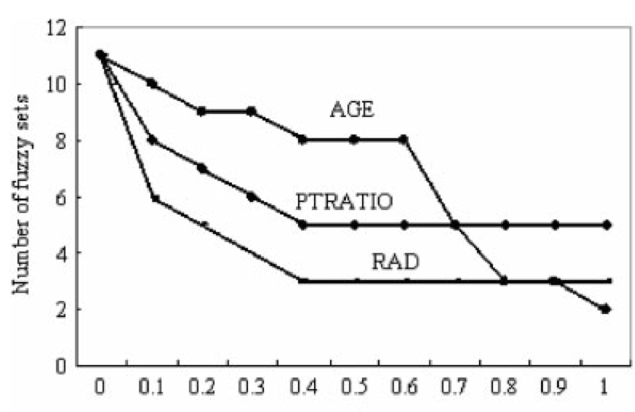

Figure 8. Number of fuzzy sets for each input variable versus ε.

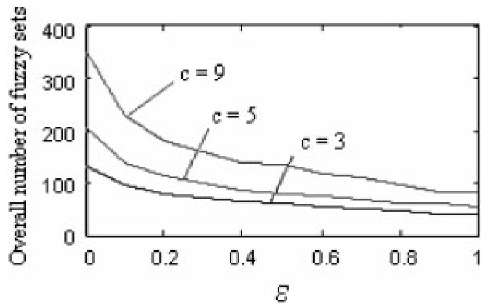

Figure 9. Overall number of fuzzy sets for inputs with three, five, and nine clusters.

(4) NOX = {VERY LOW, LOW, MEDIUM, HIGH, VERY HIGH} = {VL, L, M, H, VH}
(5) RM = {SMALL, MEDIUM, LARGE, VERY LARGE} = {S, M, L, VL}
(6) AGE = {OLD, MEDIUM, NEW} = {O, M, N}
(7) DIS = {NEAR, FAR} = {N, F}
(8) RAD = {SMALL, MEDIUM, LARGE} = {S, M, L}
(9) TAX = {VERY LOW, LOW, MEDIUM, HIGH, VERY HIGH} = {VL, L, M, H, VH}
(10) PTRATIO = {VERY LOW, LOW, MEDIUM, HIGH, VERY HIGH} = {VL, L, M, H, VH}
(11) B = {LOW, MEDIUM, HIGH} = {L, M, H}
(12) LSTAT = {LOW, MEDIUM, HIGH, VERY HIGH} = {L, M, H, VH}

With these structural components in place, we complete the gradient-based learning. For illustration, the resulting network for the LOW price is summarized in Table III. Rules are organized in term of Equations 3, 5, and 14. Because of the logic transparency of the networks, the meaning of logic description of the data is quite straightforward: low MEDV has a strong association with average room number (RM), pupil-teacher ratio (PTRATIO), population (LSTAT), and accessibility to radial highways (RAD). Some other inputs such as per capita crime rate (CRIM) and built year of houses (AGE) are less essential. For instance, in Rule 3, we

Table III. The interpretation of the network for MEDV = L.

	Context: MEDV = LOW
Performance index (RMSE)	Train = 0.1862, test = 0.2442
Rules	[(RM is M)$_{0.00}$ and (RAD is M)$_{0.00}$ and (AGE is N)$_{0.28}$ and (TAX is M)$_{0.33}$ and (NOX is M)$_{0.50}$ and (LSTAT is H)$_{0.71}$]$_{1.00}$ OR [(PTRATIO is H)$_{0.00}$ and (CRIM is M)$_{0.24}$ and (LSTAT is H)$_{0.58}$ and (AGE is N)$_{0.85}$ and (DIS is N)$_{0.92}$ and (NOX is H)$_{0.98}$]$_{1.00}$ OR [(LSTAT is H)$_{0.00}$ and (AGE is N)$_{0.34}$ and (INDUS is H)$_{0.38}$ and (ZN is L)$_{0.65}$ and (NOX is H)$_{0.90}$]$_{1.00}$

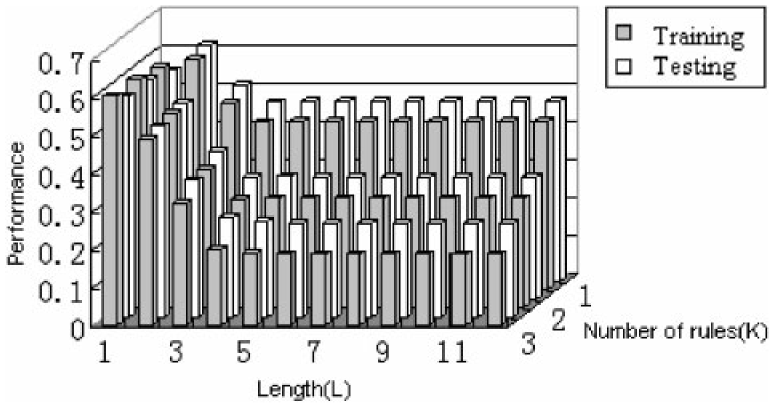

Figure 10. Values of the performance (training and testing set) treated as a function of K and L.

can note that the higher LSTAT implies lower price. Also, we can see that the proportion of blacks (B) is always weighted quite high, implying that it has no effect to the low MEDV. The rest of the expression can be interpreted in a similar manner.

Possible trade-offs between accuracy and compactness of the logic description are achieved by analyzing the values of the performance index while reducing the model and retaining a certain number of rules (K) while keeping a limited number of the conditions (L). The results shown in Figure 10 indicate that there are some values of these parameters at which the performance index does not increase while the structure has been reduced.

By inspecting the changes in the values of the performance index, we can choose the most three important rules and no more than four conditions to interpret the network. This produces the following description of the data:

Context:
MEDV = LOW
Rules:
[(RM is M)$_{0.00}$ and (RAD is M)$_{0.00}$ and (AGE is N)$_{0.28}$ and (TAX is M)$_{0.33}$]$_{1.00}$
OR
[(PTRATIO is H)$_{0.00}$ and (CRIM is M)$_{0.24}$ and (LSTAT is H)$_{0.58}$ and (AGE is N)$_{0.85}$]$_{1.00}$
OR
[(LSTAT is H)$_{0.00}$ and (AGE is N)$_{0.34}$ and (INDUS is H)$_{0.38}$ and (ZN is L)$_{0.65}$]$_{1.00}$

In a similar way, we interpret the model for the two other contexts. By considering the accuracy and compactness of the logic expressions, we choose the essential subsets of conditions and rules. Table IV summarizes two of the most important rules for each context with at most four conditions in each rule.

From Table IV, we note that real estate of medium price is characterized by medium average number of rooms (RM), comes with newer houses (AGE), and medium status of the population (LSTAT). High MEDV, low crime rates (CRIM),

Table IV. The interpretations of networks for MEDV = M and H.

	Context: MEDV = MEDIUM
Performance index (RMSE)	Train = 0.2843, test = 0.3041
Rules	$[(\text{RM is M})_{0.00}$ and $(\text{AGE is N})_{0.00}$ and $(\text{PTRATIO is M})_{0.02}$ and $(\text{CRIM is L})_{0.20}]_{1.00}$ OR $[(\text{LSTAT is M})_{0.00}$ and $(\text{RM is M})_{0.20}$ and $(\text{PTRATIO is H})_{0.22}$ and $(\text{AGE is N})_{0.59}]_{0.79}$
	Context: MEDV = HIGH
Performance index (RMSE)	Train = 0.3014, test = 0.3058
Rule	$[(\text{RM is L})_{0.00}$ and $(\text{CRIM is L})_{0.21}$ and $(\text{RAD is M})_{0.79}$ and $(\text{ZN is M})_{0.85}]_{1.00}$ OR $[(\text{CRIM is L})_{0.00}$ and $(\text{NOX is L})_{0.22}$ and $(\text{RM is L})_{0.59}$ and $(\text{PTRATIO is L})_{0.64}]_{1.00}$

larger average number of rooms (RM), and lower nitric oxides concentration (NOX) are the variables that reflect high prices.

5.2. Auto-MPG Data Set

This experimental data set comes from the UCI Machine Learning repository and deals with the fuel efficiency of various cars expressed in miles per gallon (MPG). It has six input variables: number of cylinders (CYL), displacement (DIS), horsepower (HP), weight (W), acceleration (ACC), and the model year (MODEL). As before, the fuzzy equalization was completed for three fuzzy sets, MPG ={SMALL, MEDIUM, LARGE} = {S, M, L}; see Figure 11.

Applying context-based clustering (with three clusters per context), projection, and reduction ($\varepsilon = 0.9$), we end up with the following linguistic terms formed for each input variable:

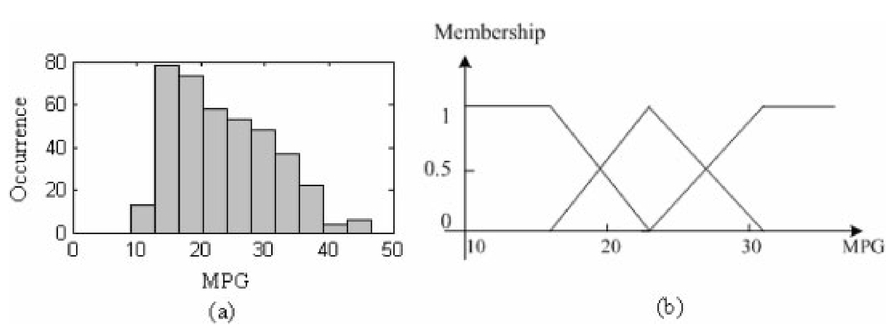

Figure 11. Fuzzy equalization of the MPG output: (a) histogram and (b) resulting fuzzy sets.

(1) CYL = {SMALL, MEDIUM, LARGE, VERY LARGE} = {S, M, L, VL}
(2) DIS = {VERY SMALL, SMALL, MEDIUM, LARGE, VERY LARGE} = {VS, S, M, L, VL}
(3) HP = {VERY SMALL, SMALL, MEDIUM, LARGE, VERY LARGE} = {VS, S, M, L, VL}
(4) W = {VERY LIGHT, LIGHT, MEDIUM, HEAVY, VERY HEAVY, EXTREMELY HEAVY} = {VL, L, M, H, VH, EH}
(5) ACC = {SMALL, MEDIUM, LARGE} = {S, M, L}
(6) MODEL = {VERY OLD, OLD, MEDIUM, NEW, VERY NEW} = {VO, O, M, N, VN}

The constructed networks come with the interpretation given in Table V.

From Table V, we note that in general vehicles with the larger number of cylinders (CYL), older models (MODEL), and higher horsepower (HP) come with lower fuel efficiency. Likewise, cars with smaller displacement (DIS) and newer built models (MODEL) are characterized by higher fuel efficiency. Medium acceleration (ACC) is strongly linked with medium fuel consumption. Thus, the ruleset that our method evolves agrees with basic intuition about the relationship between MPG and the input variables.

Table V. Interpretation of the network for MPG = {S, M, L}.

	Context: MPG = S
Performance index (RMSE)	Train = 0.1579, test = 0.1669
Rule	$[(\text{HP is L})_{0.00}$ and $(\text{CYL is VL})_{0.68}]_{1.00}$ OR $[(\text{CYL is VL})_{0.00}$ and $(\text{MODEL is O})_{0.74}$ and $(\text{HP is M})_{0.88}$ and $(\text{W is H})_{0.89}]_{1.00}$ OR $[(\text{CYL is L})_{0.09}$ and $(\text{MODEL is O})_{0.51}$ and $(\text{DIS is S})_{0.52}$ and $(\text{W is M})_{0.86}]_{1.00}$
	Context: MPG = M
Performance index (RMSE)	Train = 0.2098, test = 0.2356
Rule	$[(\text{W is M})_{0.21}$ and $(\text{MODEL is M})_{0.35}$ and $(\text{HP is S})_{0.60}$ and $(\text{ACC is M})_{0.77}]_{0.71}$ OR $[(\text{HP is S})_{0.00}$ and $(\text{ACC is M})_{0.00}$ and $(\text{W is L})_{0.01}$ and $(\text{MODEL is M})_{0.89}]_{0.70}$ OR $[(\text{DIS is VS})_{0.00}$ and $(\text{MODEL is O})_{0.00}$ and $(\text{HP is S})_{0.27}$ and $(\text{CYL is M})_{0.70}]_{0.59}$
	Context: MPG = L
Performance index (RMSE)	Train = 0.2158, test = 0.2226
Rule	$[(\text{DIS is VS})_{0.00}$ and $(\text{MODEL is N})_{0.06}$ and $(\text{CYL is M})_{0.80}]_{0.92}$ OR $[(\text{CYL is M})_{0.00}$ and $(\text{DIS is VS})_{0.00}$ and $(\text{W is VL})_{0.18}$ and $(\text{MODEL is M})_{0.69}$ and $(\text{ACC is M})_{0.92}]_{0.92}$

5.3. Computer Data Set

This data set deals with relative CPU performance, described in terms of the following attributes:

(1) MYCT: machine cycle time in nanoseconds
(2) MMIN: minimum main memory in kilobytes
(3) MMAX: maximum main memory in kilobytes
(4) CACHE: cache memory in kilobytes
(5) CHMIN: minimum channels in units
(6) CHMAX: maximum channels in units (integer)
(7) PRP: published relative performance (integer)

The performance of the CPU is quantified in terms of three contexts, PRP = {LOW, MEDIUM, HIGH} = {L, M, H}. Their design follows the standard scheme used in the previous examples; see Figure 12.

Carrying out the context-based clustering (with three clusters per cluster), projection, and reduction (here $\varepsilon = 0.3$), the list of linguistic terms is

(1) MYCT = {VERY LOW, LOW, MEDIUM, HIGH} = {VL, L, M, H}
(2) MMIN = {VERY LOW, LOW, MEDIUM, LARGE, VERY LARGE} = {VLW, LW, M, LG, VLG}
(3) MMAX = {EXTREMELY LOW, VERY LOW, LOW, MEDIUM, LARGE, VERY LARGE, EXTREMELY LARGE} = {ELW, VLW, LW, M, LG, VLG, ELG}
(4) CACHE = {VERY LOW, LOW, MEDIUM, LARGE, VERY LARGE} = {VLW, LW, M, LG, VLG}
(5) CHMIN = {VERY SMALL, SMALL, MEDIUM, LARGE, VERY LARGE} = {VS, S, M, L, VL}
(6) CHMAX = {VERY SMALL, SMALL, MEDIUM, LARGE} = {VS, S, M, L}

The rule-based description of the data is shown in Table VI.

Low maximum main memory (MMAX), low cache memory (CACHE), and small number of maximum channels (CHMAX) imply low CPU performance. With the increase of MMAX and small machine cycle time (MYCT), the performance PRP is enhanced.

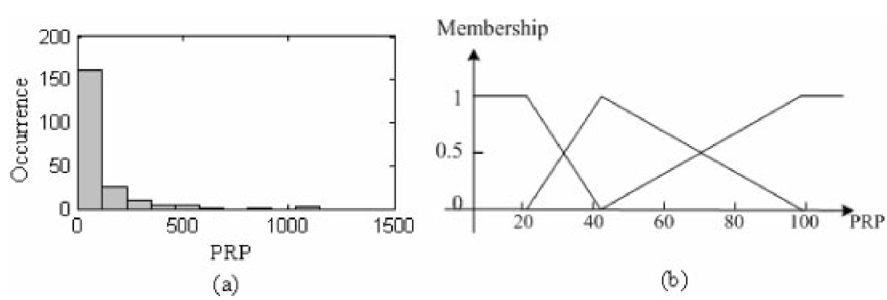

Figure 12. Fuzzy equalization of output PRP: (a) a histogram of PRP and (b) three linguistic labels.

Table VI. The interpretations of networks for PRP = {LOW, MEDIUM, HIGH}.

	Context: PRP = LOW
Performance index	Train = 0.2169, Test = 0.2402
Rule	$[(\text{MMAX is ELW})_{0.00}$ and $(\text{CACHE is VLW})_{0.00}$ and $(\text{CHMAX is VS})_{0.00}$ and $(\text{MMIN is VLW})_{0.07}$ and $(\text{CHMIN is VS})_{0.31}]_{1.00}$ OR $[(\text{MMAX is LW})_{0.00}$ and $(\text{CACHE is VLW})_{0.00}$ and $(\text{CHMAX is VS})_{0.00}$ and $(\text{MYCT is M})_{0.24}]_{1.00}$ OR $[(\text{CACHE is VLW})_{0.01}$ and $(\text{MMAX is VLW})_{0.04}$ and $(\text{CHMAX is VS})_{0.56}$ and $(\text{CHMIN is VS})_{0.88}]_{0.56}$
	Context: PRP = MEDIUM
Performance index	Train = 0.2735, test = 0.2818
Rule	$[(\text{MMAX is LG})_{0.00}$ and $(\text{CACHE is VLW})_{0.00}$ and $(\text{CHMIN is VS})_{0.10}$ and $(\text{MYCT is VL})_{0.52}$ and $(\text{MMIN is VLW})_{0.98}]_{0.65}$ OR $[(\text{MMIN is VLM})_{0.00}$ and $(\text{MYCT is L})_{0.34}$ and $(\text{MMAX is VLW})_{0.82}]_{0.61}$ OR $[(\text{MYCT is VL})_{0.00}$ and $(\text{CHMAX is VS})_{0.00}$ and $(\text{MMAX is LW})_{0.15}$ and $(\text{CACHE is VLW})_{0.62}]_{0.51}$
	Context: PRP = HIGH
Performance index	Train = 0.2726, test = 0.2898
Rule	$[(\text{MMAX is ELG})_{0.00}$ and $(\text{MYCT is VL})_{0.22}]_{1.00}$ OR $[(\text{MYCT is VL})_{0.08}$ and $(\text{CHMIN is S})_{0.52}$ and $(\text{MMIN is LW})_{0.58}$ and $(\text{CACHE is LW})_{0.96}]_{1.00}$

6. CONCLUSIONS

This study describes a comprehensive set of experiments about fuzzy logic-based neural networks and their interpretative abilities. The proposed fuzzy neural network exhibits transparent logic interpretation as well as significant learning abilities. We also explored the use of fuzzy equalization, conditional FCM, and gradient-based learning to realize structural and parametric optimization of the network.

The article covers a number of experimental studies that offer some insight into the performance of the logic descriptions and interpretation abilities. This extensive suite of experiments, including several that were not presented here, leads us to some general observations. The performance of the network on the training and testing set expressed as the ratio of Q_{test}/Q_{train} varies from 101.46% to 131.15%, so, on average, performance on the test sets deteriorated by 30.86%. The networks led to a fairly consistent logic description of the experimental data resulting in 2.33 rules (on average) with an average length of 4.06 variables. Interestingly, each model used only a portion of all inputs (viz., 27.27%–83.33% of all inputs). This is quite indicative of the redundancy existing in the data; outputs can usually be quite well described by a limited number of system inputs.

Acknowledgment

Support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Canada Research Chair (CRC) program is gratefully acknowledged.

References

1. Lee SC, Lee ET. Fuzzy sets and neural networks. J Cybern 1974;4:83–103.
2. Lee SC, Lee ET. Fuzzy neural networks. Math Biosci 1975;23:151–177.
3. Chung FL, Duan JC. On multistage fuzzy neural network modeling. IEEE Trans Fuzzy Syst 2000;8:125–142.
4. Horikawa S, Furuhashi T, Uchikawa Y. On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm. IEEE Trans Neural Netw 1992;3(5):801–806.
5. Jang J-SR. ANFIS: Adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 1993;23:665–685.
6. Kosko B. Neural networks and fuzzy systems. Englewood Cliffs, NJ: Prentice-Hall; 1991.
7. Mitra S, Pal SK. Logical operation based fuzzy MLP for classification and rule generation. Neural Netw 1994;7:353–373.
8. Mitra S, Pal SK. Fuzzy multiplayer perceptron, inferencing, and rule generation. IEEE Trans Neural Netw 1995;6:51–63.
9. Pedrycz W. Fuzzy neural networks and neurocomputations. Fuzzy Set Syst 1993;56:1–28.
10. Pedrycz W, Gomide F. An introduction to fuzzy sets: Analysis and design. Cambridge, MA: MIT Press; 1998.
11. Hirota K, Pedrycz W. OR/AND neuron in modeling fuzzy set connectives. IEEE Trans Fuzzy Syst 1994;2:151–161.
12. Pedrycz W. Heterogeneous fuzzy logic networks: Fundamentals and development studies. IEEE Trans Neural Netw 2004;15:1466–1481.
13. Pedrycz W. Fuzzy sets engineering. Boca Raton, FL: CRC Press; 1995.
14. Pedrycz W, Succi G. fXOR fuzzy logic networks. Soft Comput 2002;7:115–120.
15. Pedrycz W. Fuzzy equalization in the construction of fuzzy sets. Fuzzy Set Syst 2001; 119:329–335.
16. Pedrycz W. Conditional Fuzzy C—Means. Pattern Recogn Lett 1996;17:625–632.
17. Pedrycz W. Conditional fuzzy clustering in the design of radial basis function neural networks. IEEE Trans Neural Netw 1998;9:601–612.
18. Bezdek JC. Pattern recognition with fuzzy objective function algorithms. New York: Plenum; 1981.