Information Sciences 179 (2009) 2175–2193

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks

Juan R. Castro [a], Oscar Castillo [b,*], Patricia Melin [b], Antonio Rodríguez-Díaz [a]

[a] UABC University, Tijuana, Mexico
[b] Tijuana Institute of Technology, Division of Graduate Studies and Research, Department of Computer Science, 22500 Tijuana, Mexico

ARTICLE INFO

*Article history:*
Received 8 November 2007
Received in revised form 19 August 2008
Accepted 4 October 2008

*Keywords:*
Interval type-2 fuzzy neural networks
Interval type-2 fuzzy neuron
Hybrid learning algorithm
Interval type-2 fuzzy inference systems

ABSTRACT

In real life, information about the world is uncertain and imprecise. The cause of this uncertainty is due to: deficiencies on given information, the fuzzy nature of our perception of events and objects, and on the limitations of the models we use to explain the world. The development of new methods for dealing with information with uncertainty is crucial for solving real life problems. In this paper three interval type-2 fuzzy neural network (IT2FNN) architectures are proposed, with hybrid learning algorithm techniques (gradient descent backpropagation and gradient descent with adaptive learning rate backpropagation). At the antecedents layer, a interval type-2 fuzzy neuron (IT2FN) model is used, and in case of the consequents layer an interval type-1 fuzzy neuron model (IT1FN), in order to fuzzify the rule's antecedents and consequents of an interval type-2 Takagi–Sugeno–Kang fuzzy inference system (IT2-TSK-FIS). IT2-TSK-FIS is integrated in an adaptive neural network, in order to take advantage the best of both models. This provides a high order intuitive mechanism for representing imperfect information by means of use of fuzzy If–Then rules, in addition to handling uncertainty and imprecision. On the other hand, neural networks are highly adaptable, with learning and generalization capabilities. Experimental results are divided in two kinds: in the first one a non-linear identification problem for control systems is simulated, here a comparative analysis of learning architectures IT2FNN and ANFIS is done. For the second kind, a non-linear Mackey–Glass chaotic time series prediction problem with uncertainty sources is studied. Finally, IT2FNN proved to be more efficient mechanism for modeling real-world problems.



## 1. Introduction

Hybrid intelligent systems combining fuzzy logic (FL), neural networks (NN), genetic algorithms (GA), and expert systems (ES) are proving their effectiveness in a wide variety of real-world problems [1,22–25]. Every intelligent technique has particular computational properties (e.g. ability to learn and explanation of decisions) that make them suitable for special kinds of problems and not for others. For example, while neural networks are good at recognizing patterns, they are not good at explaining how they reach their decisions. Fuzzy systems [2,20,11–13], which can reason with imprecise information and uncertainty, are good at explaining their decisions but they cannot automatically acquire the rules they use to make those decisions. These limitations have been a central driving force behind the creation of intelligent hybrid systems, where two or more techniques are combined in a manner that overcomes the limitations of individual techniques. Intelligent hybrid systems are also important when considering the varied nature of application domains. Many complex domains have many dif-

* Corresponding author.
*E-mail address:* ocastillo@hafsamx.org (O. Castillo).

0020-0255/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.10.016

ferent component problems, each of which may require different types of processing. If there is a complex application, which has two distinct sub-problems (e.g. a signal processing task and a serial reasoning task), then a neural network and an expert system, respectively can be used for solving these separate tasks. The use of intelligent hybrid systems is growing rapidly with successful applications in many areas including process control, engineering design, financial trading, credit evaluation, medical diagnosis, and cognitive simulation [1,21,28,30,31,36,37].

While interval type-2 fuzzy logic (IT2FL) provides an inference mechanism under cognitive uncertainty, computational neural networks offer exciting advantages, such as learning, adaptation, fault-tolerance and generalization [32,33]. To enable a system to deal with cognitive uncertainties in a manner more like humans do, one may incorporate the concept of interval type-2 fuzzy logic into neural networks.

The idea of considering type-1 fuzzy neurons (T1FN) was introduced by Pedrycz [3], Pedrycz and Rocha [4]. While the models of those neurons involve *max* and *min* operators and triangular norms, the neuron presented in this paper utilizes a particular extension of fuzzy sets [38,39].

The computational process envisioned for fuzzy neural systems is as follows: it starts with the development of an "interval type-2 fuzzy neuron (IT2FN)" based on the understanding of biological neural morphologies, followed by the learning mechanisms. This leads to the following three steps in an interval type-2 fuzzy neural computational process:

- Development of interval type-2 fuzzy neural models motivated by biological neurons.
- Models of synaptic connections, which incorporate fuzziness into neural networks.
- Development of learning algorithms (that is, the method of adjusting the synaptic weights).

An interval type-2 fuzzy neural network (IT2FNN) is a neural network with interval type-2 fuzzy signals and/or interval type-2 fuzzy weights, Gaussian, generalized bell and sigmoid transfer function, and all operations defined by Zadeh's [5–10] extension's principle.

## 2. Interval type-2 fuzzy logic systems

An Interval type-2 fuzzy set can represent and handle uncertain information effectively. That is, interval type-2 fuzzy sets let us model and minimizes the effects of uncertainties in rule-base interval type-2 fuzzy logic systems (IT2FLS). A general interval type-2 fuzzy logic system is depicted in Fig. 1. An IT2FLS is very similar to a type-1 fuzzy logic systems (FLS) [1,2,35] the major structural difference being that the defuzzifier block of type-1 FLS is replaced by the output processing block in an interval type-2 FLS, which consists of type-reduction followed by defuzzification.

Consider an interval type-2 Takagi–Sugeno–Kang (TSK) FLS having $n$ inputs $x_1 \in X_1, \ldots, x_i \in X_i, \ldots, x_n \in X_n$ and $m$ outputs $y_1 \in Y_1, \ldots, y_j \in Y_j, \ldots, y_m \in Y_m$. An interval type-2 TSK FLS is also described by fuzzy If–Then rules that represent input–output relations of a system. In general, a first-order interval type-2 TSK [1,15,16,30,31,34] models with a rule base of $M$ rules, each having $n$ antecedents, the $k$th rule can be expressed as follows:

$$\begin{aligned} &R^k : \text{IF } x_1 \text{ is } \widetilde{A}_1^k \text{ and } \cdots \text{ and } x_i \text{ is } \widetilde{A}_i^k \text{ and } \cdots \text{ and } x_n \text{ is } \widetilde{A}_n^k \\ &\text{THEN } y_j \text{ is } \tilde{y}_k^j = C_{k,1}^j x_1 + \cdots + C_{k,i}^j x_i + \cdots + C_{k,n}^j x_n + C_{k,0}^j, \end{aligned}$$

where $k = 1, \ldots, M, C_{k,i}^j$ ($i$=0,1,...,$n$; $j$=1,...,$m$) are consequent interval type-1 fuzzy sets; $\tilde{y}_k^j$. The $j$th output of the $k$th rule is also an interval type-1 fuzzy set (since it is a linear combination of interval type-1 fuzzy sets); and $\widetilde{A}_i^k$ $(i = 1, \ldots, n)$ are interval type-2 antecedent fuzzy sets. These rules take into account simultaneously uncertainty about antecedent membership functions and consequent parameter values.

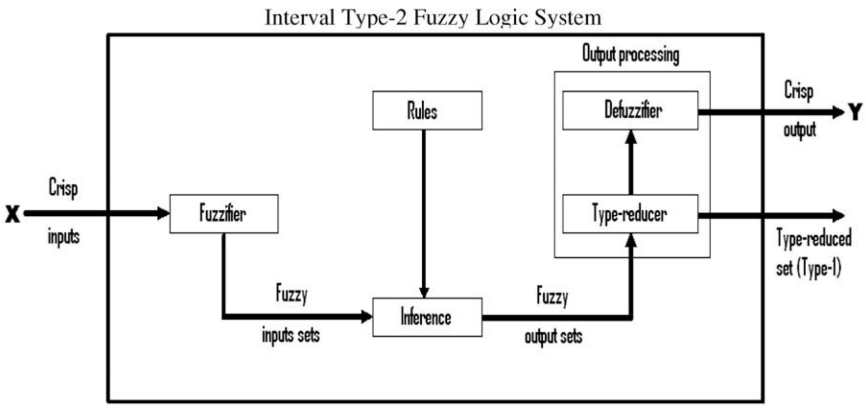


**Fig. 1.** The block diagram of type-2 fuzzy logic system.

In an interval type-2 TSK FLS with *meet* $\sqcap$ under product or minimum *t*-norm [13], the firing set of the *k*th rule is $F^k(x)$, which is an interval type-1 set defined as

$$F^k(x) = \prod_{i=1}^{n} \mu_{\tilde{A}_i^k}(x_i) = [\underline{f}^k(x), \bar{f}^k(x)], \tag{1}$$

where

$$\underline{f}^k(x) = \mathop{\tilde{*}}_{i=1}^{n} [\underline{\mu}_{\tilde{A}_i^k}(x_i)] \quad \text{and} \quad \bar{f}^k(x) = \mathop{\tilde{*}}_{i=1}^{n} [\bar{\mu}_{\tilde{A}_i^k}(x_i)]. \tag{2}$$

The consequent of rule $R^k, \tilde{y}_k^j = [{}_l y_k^j, {}_r y_k^j]$, is also an interval; $\tilde{y}_k^j = C_{k,1}^j x_1 + \cdots + C_{k,i}^j x_i + \cdots + C_{k,n}^j x_n + C_{k,0}^j$ set; and $C_{k,i}^j \in [c_{k,i}^j - s_{k,i}^j, c_{k,i}^j + s_{k,i}^j]$ where $c_{k,i}^j$ denotes the center (mean) of $C_{k,i}^j$, $s_{k,i}^j$ denotes the spread of $C_{k,i}^j$ and ${}_l y_k^j, y_k^j$ are defined by

$$\begin{aligned} {}_l y_k^j &= \sum_{i=1}^{n} c_{k,i}^j x_i + c_{k,0}^j - \sum_{i=1}^{n} s_{k,i}^j |x_i| - s_{k,0}^j, \\ {}_r y_k^j &= \sum_{i=1}^{n} c_{k,i}^j x_i + c_{k,0}^j + \sum_{i=1}^{n} s_{k,i}^j |x_i| + s_{k,0}^j. \end{aligned} \tag{3}$$

The *j*th output of a first-order interval type-2 TSK FLS is obtained by applying the extension principle, where now both $f^k(x)$ and $y_k^j(x)$ are replaced by interval type-1 fuzzy sets. Hence, $\widetilde{Y}^j_{\mathrm{TSK},2}(x)$ is an interval type-1 set. To compute $\widetilde{Y}^j_{\mathrm{TSK},2}(x)$, we therefore only need to compute its two end-points ${}_l y^j$ and ${}_r y^j$ as follows:

$$\widetilde{Y}^j_{\mathrm{TSK},2}(x) = [{}_l y^j, {}_r y^j] = \int_{y_1^j \in [{}_l y_1^j, {}_r y_1^j]} \cdots \int_{y_M^j \in [{}_l y_M^j, {}_r y_M^j]} \int_{f^1 \in [f_l^1, f_r^1]} \cdots \int_{f^M \in [f_l^M, f_r^M]} 1 \Bigg/ \frac{\sum_{k=1}^{M} f^k y_k^j}{\sum_{k=1}^{M} f^k}. \tag{4}$$

In an interval type-2 TSK FLS, $\widetilde{Y}^j_{\mathrm{TSK},2}(x)$ is an interval type-1 fuzzy set. $\widetilde{Y}^j_{\mathrm{TSK},2}(x)$ is defuzzified using the average of ${}_l y^j$ and ${}_r y^j$; hence, the defuzzified output of any interval type-2 TSK FLS is

$$y^j_{\mathrm{TSK},2}(x) = \frac{{}_l y^j + {}_r y^j}{2}. \tag{5}$$

For example, Fig. 2 shows an interval type-2 TSK FLS [14,26] with two inputs ($x_1$,$x_2$), one output ($y_1$) and four rules ($k = 1,\ldots,4$); each input fuzzified by four membership functions ($A_{1,k}$,$A_{2,k}$) and each output is fuzzified by four interval linear membership functions $f_{1,k} \in (y_l^k, y_r^k)$.

## 3. Interval type-2 fuzzy neural networks

One way to build interval type-2 fuzzy neural networks (IT2FNN) is to fuzzify a conventional neural network. Each part of a neural network (the activation function, the weights, and the inputs and outputs) can be fuzzified. A fuzzy neuron is basically similar to an artificial neuron, except that it has the ability to process fuzzy information.

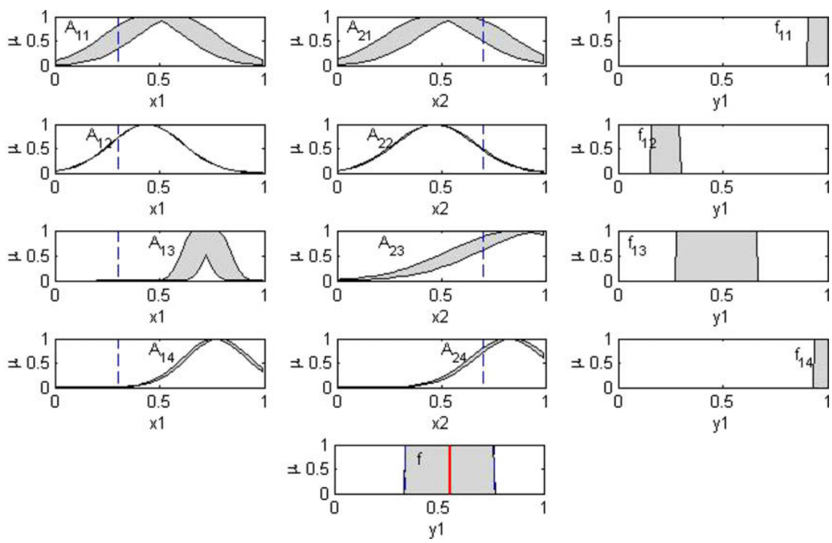


**Fig. 2.** Interval type-2 TSK FLS.

The interval type-2 fuzzy neural network (IT2FNN) system is one kind of interval Takagi–Sugeno–Kang fuzzy inference system (IT2-TSK-FIS) in neural network structure. An IT2FNN is proposed, with TSK reasoning and processing elements called interval type-2 fuzzy neurons (IT2FN) for defining antecedents, and interval type-1 fuzzy neurons (IT1FN) for defining the consequents of rules $R^k$.

An IT2FN is composed by two adaptive nodes represented by squares, and two non-adaptive nodes represented by circles. Adaptive nodes have outputs that depend on their inputs, modifiable parameters and transference function while non-adaptive, on the contrary, depend solely on their inputs, and their outputs represent lower ($\underline{\mu}(x)$) and upper ($\bar{\mu}(x)$) membership functions. Parameters from adaptive nodes with uncertain standard deviation are denoted by ($w \in [w_{1,1}, w_{2,1}]$) and with uncertain mean by ($b \in [b_1, b_2]$). Fig. 3 shows an IT2FN with crisp input signals ($x$), crisp synaptic weights ($w, b$) and type-1 fuzzy outputs $\mu(net_1), \mu(net_2), \underline{\mu}(x), \bar{\mu}(x)$. This kind of neuron is build from two conventional neurons with transference functions $\mu(net)$, Gaussian, generalized bell and logistic for fuzzifier the inputs. Each neuron equation is defined as follows:

The function $\mu$ is often referred to as an activation (or transfer) function. Its domain is the set of activation values, **net**, of the neuron model; we thus often use this function as $\mu(net)$. The variable **net** is defined as a scalar product of the weight and input vectors:

$$\begin{aligned} net_1 &= w_{1,1}x + b_1; \quad \mu_1 = \mu(net_1), \\ net_2 &= w_{2,1}x + b_1; \quad \mu_2 = \mu(net_2). \end{aligned} \tag{6}$$

Non-adaptive node $t$-norm ($T$) evaluates lower membership function ($\underline{\mu}(x)$) under algebraic product $t$-norm while non-adaptive node $s$-norm ($S$) evaluates upper membership function ($\bar{\mu}(x)$) under algebraic sum $s$-norm, as shown in Eq. (7):

$$\begin{aligned} \underline{\mu}(x) &= \mu(net_1) \cdot \mu(net_2), \\ \bar{\mu}(x) &= \mu(net_1) + \mu(net_1) - \underline{\mu}(x). \end{aligned} \tag{7}$$

Each IT2FN adapts an interval type-2 fuzzy set [2], $\widetilde{A}$, expressed in terms of output $\underline{\mu}(x)$, of type-1 fuzzy neuron with $T$-norm and $\bar{\mu}(x)$ of type-1 fuzzy neuron with $S$-norm. An interval type-2 fuzzy set is denoted as

$$\widetilde{A} = \int_{x \in X} \left[ \int_{\mu(x) \in [\underline{\mu}(x), \bar{\mu}(x)]} 1/\mu \right] \Big/ x. \tag{8}$$

Fig. 4 shows an interval type-1 fuzzy neuron build from two conventional adaptive linear neurons (ADALINE) [29] for adapting consequents $\tilde{y}_k^j \in [{}_l y_k^j, {}_r y_k^j]$ from rules $R^k$, for the $j$th output defined by

$$\begin{aligned} {}_l y_k^j &= \sum_{i=1}^{n} c_{k,i}^j x_i + c_{k,0}^j - \sum_{i=1}^{n} s_{k,i}^j |x_i| - s_{k,0}^j, \\ {}_r y_k^j &= \sum_{i=1}^{n} c_{k,i}^j x_i + c_{k,0}^j + \sum_{i=1}^{n} s_{k,i}^j |x_i| + s_{k,0}^j. \end{aligned} \tag{9}$$

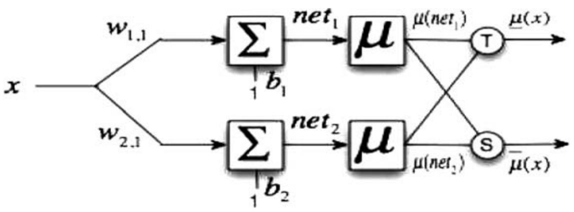


**Fig. 3.** Interval type-2 fuzzy neuron (IT2FN).

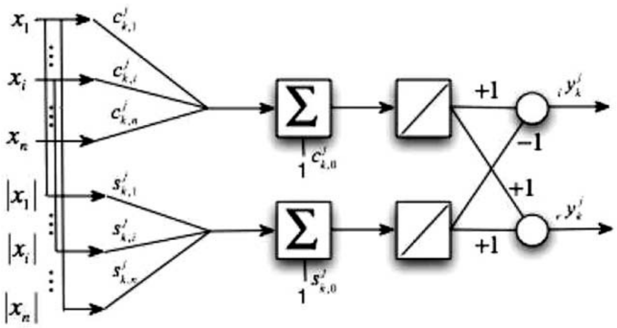


**Fig. 4.** Interval type-1 fuzzy neuron.

Thus, consequents can be adapted with a network of adaptive linear networks (MADALINE) [29]. The network weights are the parameters of consequents $c^j_{k,i}$ and $s^j_{k,i}$ for the $k$th rule. The outputs represent interval linear membership functions of the rule's consequents, as shown in Fig. 5.

In this paper, three new IT2FNN architectures are proposed for integrating a type TSK-IT2FIS in an adaptive neural network. Main differences between all architectures are: the way antecedents (adaptive nodes or interval type-2 fuzzy neurons) and consequents (linear adaptive nodes or interval type-1 fuzzy neurons) are integrated, and the way reduction type in the neural network layers is integrated. First architecture uses only adaptive nodes (does not utilize IT2FN and IT1FN), and utilizes Karnik–Mendel algorithm for type-reduction. Second architecture uses IT2FN, IT1FN and Karnik–Mendel algorithm. Third architecture is similar to second except reduction type is integrated in the adaptive network layers.

First architecture IT2FNN is an adaptive neural network with five layers (Fig. 6). Layers 0 and 5 represent inputs and outputs, respectively. Layer 1 contains adaptive nodes with transference function equivalent to intercalated lower–upper membership functions, which are semi-continuos and differentiable functions. Layer 2 contains non-adaptive nodes that represent rules antecedents and each output represents lower–upper firing set of a rule. Layer 3 contains adaptive nodes the represent consequents of each rule. Layer 4 evaluates type-reduction using Karnik–Mendel algorithm [2]. Layer five defuzzifies the system's output.

For simplicity, we assume the IT2FNN under consideration has $\mathbf{n}$ inputs and one output. The forward-propagation procedure is described as follows:

*Layer 0*: Inputs

$$x = (x_1, \ldots, x_i, \ldots, x_n)^t.$$

*Layer 1*: Interval type-2 member functions, $\tilde{\mu}_{k,i}(x_i) = \{\underline{\mu}_{k,i}(x_i), \bar{\mu}_{k,i}(x_i)\}$.
For example:

$$\mu_{k,i}(x_i) = \{\underline{\mu}_{k,i}(x_i), \bar{\mu}_{k,i}(x_i)\} = igaussmtype2(x_i, [\sigma_{k,i}, {}^1m_{k,i}, {}^2m_{k,i}]); \quad k = 1, 2, \ldots, M; \;\; i = 1, 2, \ldots, n,$$

$${}^1\mu_{k,i}(x_i, [\sigma_{k,i}, {}^1m_{k,i}]) = e^{-1/2\left(\frac{x_i - {}^1m_{k,i}}{\sigma_{k,i}}\right)^2}; \quad {}^2\mu_{k,i}(x_i, [\sigma_{k,i}, {}^2m_{k,i}]) = e^{-1/2\left(\frac{x_i - {}^2m_{k,i}}{\sigma_{k,i}}\right)^2},$$

$$\bar{\mu}_{k,i}(x_i) = \begin{cases} {}^1\mu(x_i, [\sigma_{k,i}, {}^1m_{k,i}]), & x_i < {}^1m_{k,i}, \\ 1, & {}^1m_{k,i} \leqslant x_i \geqslant {}^2m_{k,i}, \\ {}^2\mu(x_i, [\sigma_{k,i}, {}^2m_{k,i}]), & x_i > {}^2m_{k,i}, \end{cases}$$

$$\underline{\mu}_{k,i}(x_i) = \begin{cases} {}^2\mu(x_i, [\sigma_{k,i}, {}^2m_{k,i}]), & x_i \leqslant \frac{{}^1m_{k,i} + {}^2m_{k,i}}{2}, \\ {}^1\mu(x_i, [\sigma_{k,i}, {}^1m_{k,i}]), & x_i > \frac{{}^1m_{k,i} + {}^2m_{k,i}}{2}. \end{cases}$$

*Layer 2*: Rules

$$\underline{\underline{f}^k = \tilde{\ast}_{i=1}^{n}(\underline{\mu}_{k,i})}; \quad \underline{\bar{f}^k = \tilde{\ast}_{i=1}^{n}(\bar{\mu}_{k,i})}.$$

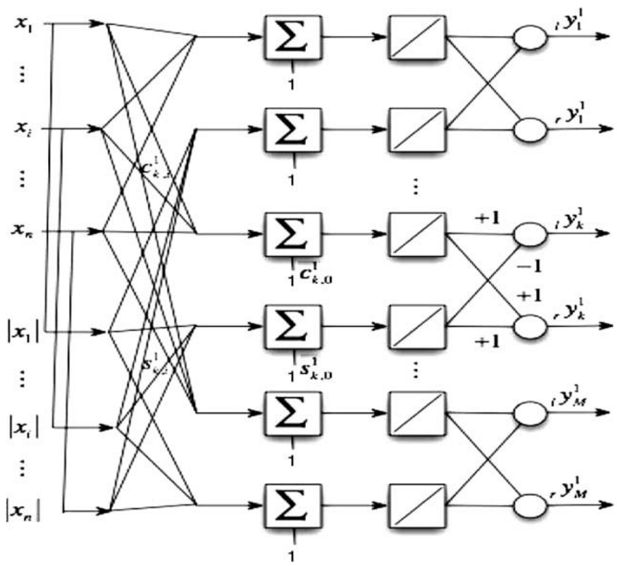


**Fig. 5.** Interval type-1 fuzzy neural network.

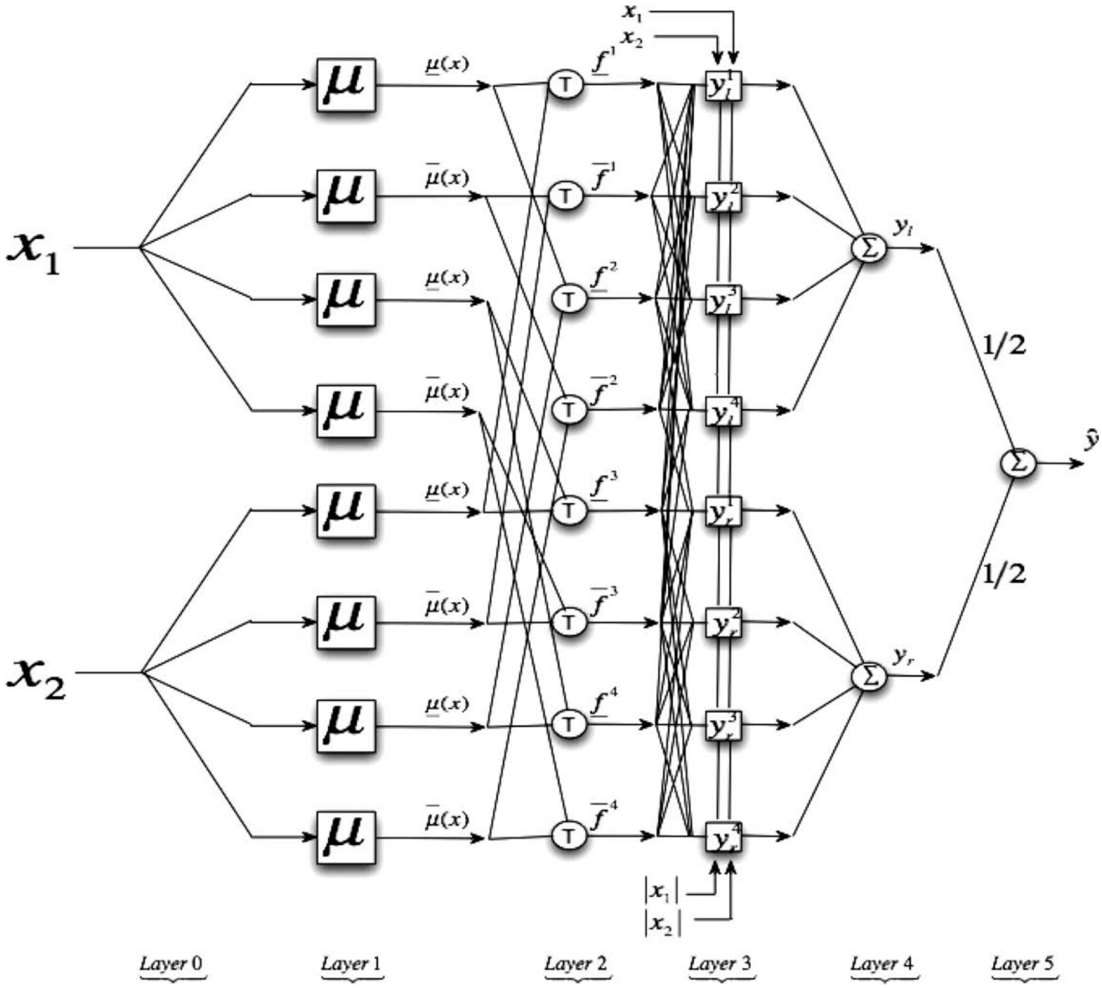


**Fig. 6.** Interval type-2 fuzzy neural network.

*Layer 3*: Consequent left–right firing points

$$y_l^k = \sum_{i=1}^{n} c_{k,i} x_i + c_{k,0} - \sum_{i=1}^{n} s_{k,i} |x_i| - s_{k,0},$$
$$y_r^k = \sum_{i=1}^{n} c_{k,i} x_i + c_{k,0} + \sum_{i=1}^{n} s_{k,i} |x_i| + s_{k,0}.$$

*Layer 4*: Left–right points (type-reduction using KM algorithm)

$$\hat{y}_l = \hat{y}_l(\bar{f}^1, \ldots, \bar{f}^L, \underline{f}^{L+1}, \ldots, \underline{f}^M, y_l^1, \ldots, y_l^M) = \frac{\sum_{k=1}^{M} f_l^k \cdot y_l^k}{\sum_{k=1}^{M} f_l^k} = \frac{\sum_{k=1}^{L} \bar{f}^k \cdot y_l^k + \sum_{k=L+1}^{M} \underline{f}^k \cdot y_l^k}{\sum_{k=1}^{L} \bar{f}^k + \sum_{k=L+1}^{M} \underline{f}^k},$$
$$\hat{y}_r = \hat{y}_r(\underline{f}^1, \ldots, \underline{f}^R, \bar{f}^{R+1}, \ldots, \bar{f}^M, y_r^1, \ldots, y_r^M) = \frac{\sum_{k=1}^{M} f_r^k \cdot y_r^k}{\sum_{k=1}^{M} f_r^k} = \frac{\sum_{k=1}^{R} \underline{f}^k \cdot y_r^k + \sum_{k=R+1}^{M} \bar{f}^k \cdot y_r^k}{\sum_{k=1}^{R} \underline{f}^k + \sum_{k=R+1}^{M} \bar{f}^k}.$$

*Layer 5*: Defuzzification

$$\hat{y} = \frac{\hat{y}_l + \hat{y}_r}{2}.$$

The IT2FNN uses backpropagation (steepest descent) method [27] for learning how to determine premise parameters (to find the parameters related to interval membership functions) and consequent parameters. The learning procedure has two parts: in the first part the input patterns are propagated, the consequent parameters and the premise parameters are assumed to be fixed for the current cycle through the training set. In the second part the patterns are propagated again, and at this moment, backpropagation is used to modify the premise parameters, and consequent parameters. These two parts are considered to be an epoch. Given an input–output training pair $\{(x_p : t_p)\}\ \forall p = 1,\dots,q$, in order to get the design of the IT2FNN, the error function ($E$) must be minimized:

$$e_p = t_p - \hat{y}_p, \tag{10}$$

$$E_p = \frac{1}{2}e_p^2 = \frac{1}{2}(t_p - \hat{y}_p)^2, \tag{11}$$

$$E = \sum_{p=1}^{q} E_p. \tag{12}$$

Accordingly, the update formulas for the generic antecedents ($\xi_{k,i}$) and consequents ($c_{k,i}, s_{k,i}$), are given by Eqs. (13)–(15), respectively

$$^{\text{new}}c_{k,i} = {}^{\text{old}}c_{k,i} - \eta \cdot \frac{\partial E_p}{\partial c_{k,i}}, \tag{13}$$

$$^{\text{new}}s_{k,i} = {}^{\text{old}}s_{k,i} - \eta \cdot \frac{\partial E_p}{\partial s_{k,i}}, \tag{14}$$

$$^{\text{new}}\xi_{k,i} = {}^{\text{old}}\xi_{k,i} - \eta \cdot \frac{\partial E_p}{\partial \xi_{k,i}}, \tag{15}$$

where $\eta$ is a learning rate.

Equations from (16)–(19) update the parameters of the consequents of the rules

$$^{\text{new}}c_{k,i} = {}^{\text{old}}c_{k,i} + \eta \cdot \frac{1}{2} \cdot e_p \left[\frac{f_r^k}{\sum_{k=1}^{M} f_r^k} + \frac{f_l^k}{\sum_{k=1}^{M} f_l^k}\right] \cdot x_i, \tag{16}$$

$$^{\text{new}}c_{k,0} = {}^{\text{old}}c_{k,0} + \eta \cdot \frac{1}{2} \cdot e_p \left[\frac{f_r^k}{\sum_{k=1}^{M} f_r^k} + \frac{f_l^k}{\sum_{k=1}^{M} f_l^k}\right], \tag{17}$$

$$^{\text{new}}s_{k,i} = {}^{\text{old}}s_{k,i} + \eta \cdot \frac{1}{2} \cdot e_p \left[\frac{f_r^k}{\sum_{k=1}^{M} f_r^k} - \frac{f_l^k}{\sum_{k=1}^{M} f_l^k}\right] \cdot |x_i|, \tag{18}$$

$$^{\text{new}}s_{k,0} = {}^{\text{old}}s_{k,0} + \eta \cdot \frac{1}{2} \cdot e_p \left[\frac{f_r^k}{\sum_{k=1}^{M} f_r^k} - \frac{f_l^k}{\sum_{k=1}^{M} f_l^k}\right]. \tag{19}$$

Equations from (20)–(23) update the parameters of the antecedents of the rules

$$\xi_{k,i}^{\text{new}} = \xi_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_l^k - \hat{y}_l}{D_l} \cdot \underset{l=1, l\neq i}{\overset{n}{\tilde{*}}}(\bar{\mu}_{k,l}) \cdot \frac{\partial \bar{\mu}_{k,i}}{\partial \xi_{k,i}}; \quad k \leqslant L, \tag{20}$$

$$\xi_{k,i}^{\text{new}} = \xi_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_l^k - \hat{y}_l}{D_l} \cdot \underset{l=1, l\neq i}{\overset{n}{\tilde{*}}}(\underline{\mu}_{k,l}) \cdot \frac{\partial \underline{\mu}_{k,i}}{\partial \xi_{k,i}}; \quad k > L, \tag{21}$$

$$\xi_{k,i}^{\text{new}} = \xi_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_r^k - \hat{y}_r}{D_r} \cdot \underset{l=1, l\neq i}{\overset{n}{\tilde{*}}}(\underline{\mu}_{k,l}) \cdot \frac{\partial \underline{\mu}_{k,i}}{\partial \xi_{k,i}}; \quad k \leqslant R, \tag{22}$$

$$\xi_{k,i}^{\text{new}} = \xi_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_r^k - \hat{y}_r}{D_r} \cdot \underset{l=1, l\neq i}{\overset{n}{\tilde{*}}}(\bar{\mu}_{k,l}) \cdot \frac{\partial \bar{\mu}_{k,i}}{\partial \xi_{k,i}}; \quad k > R \tag{23}$$

for example, if $\mu_{k,i}(x_i) = \{\underline{\mu}_{k,i}(x_i), \bar{\mu}_{k,i}(x_i)\} = igaussmtype2(x_i, [\sigma_{k,i}, {}^1m_{k,i}, {}^2m_{k,i}])$; where $k = 1,2,\dots,M$; $i = 1,2,\dots,n$ then

$k \leqslant L$:

$x_i < {}^1m_{k,i}$ (Eqs. (24) and (25) only have an effect on ${}^1\mu_{k,i}$)

$$^1m_{k,i}^{\text{new}} = {}^1m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_l^k - \hat{y}_l) \cdot \frac{\bar{f}^k}{\sum_{j=1}^{L} f_l^j} \cdot \frac{x_i - {}^1m_{k,i}}{(\sigma_{k,i})^2}, \tag{24}$$

$$\sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_l^k - \hat{y}_l) \cdot \frac{\bar{f}^k}{\sum_{j=1}^{L} f_l^j} \cdot \frac{(x_i - {}^1m_{k,i})^2}{(\sigma_{k,i})^3}. \tag{25}$$

$x_i > {}^2m_{k,i}$ (Eqs. (26) and (27) only have an effect on ${}^2\mu_{k,i}$)

$$^2m_{k,i}^{\text{new}} = {}^2m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_l^k - \hat{y}_l) \cdot \frac{\bar{f}^k}{\sum_{j=1}^{L} f_l^j} \cdot \frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2}, \tag{26}$$

$$\sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_l^k - \hat{y}_l) \cdot \frac{\bar{f}^k}{\sum_{j=1}^{L} f_l^j} \cdot \frac{(x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3}. \tag{27}$$

$k > L$:
$x_i < \frac{{}^1m_{k,i}+{}^2m_{k,i}}{2}$ (Eqs. (28) and (29) only have an effect on ${}^2\mu_{k,i}$)

$$ {}^2m_{k,i}^{\text{new}} = {}^2m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_l^k - \hat{y}_l) \cdot \frac{\underline{f}^k}{\sum_{j=L+1}^{M} \underline{f}_l^j} \cdot \frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2}, \tag{28} $$

$$ \sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_l^k - \hat{y}_l) \cdot \frac{\underline{f}^k}{\sum_{j=L+1}^{M} \underline{f}_l^j} \cdot \frac{(x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3}. \tag{29} $$

$x_i \geqslant \frac{{}^1m_{k,i}+{}^2m_{k,i}}{2}$ (Eqs. (30) and (31) only have an effect on ${}^1\mu_{k,i}$)

$$ {}^1m_{k,i}^{\text{new}} = {}^1m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_l^k - \hat{y}_l) \cdot \frac{\underline{f}^k}{\sum_{j=L+1}^{M} \underline{f}_l^j} \cdot \frac{x_i - {}^1m_{k,i}}{(\sigma_{k,i})^2}, \tag{30} $$

$$ \sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_l^k - \hat{y}_l) \cdot \frac{\underline{f}^k}{\sum_{j=L+1}^{M} \underline{f}_l^j} \cdot \frac{(x_i - {}^1m_{k,i})^2}{(\sigma_{k,i})^3}. \tag{31} $$

$k \leqslant R$:
$x_i < \frac{{}^1m_{k,i}+{}^2m_{k,i}}{2}$ (Eqs. (32) and (33) only have an effect on ${}^2\mu_{k,i}$)

$$ {}^2m_{k,i}^{\text{new}} = {}^2m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{\underline{f}^k}{\sum_{j=1}^{R} \underline{f}_r^j} \cdot \frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2}, \tag{32} $$

$$ \sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{\underline{f}^k}{\sum_{j=1}^{R} \underline{f}_r^j} \cdot \frac{(x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3}. \tag{33} $$

$x_i \geqslant \frac{{}^1m_{k,i}+{}^2m_{k,i}}{2}$ (Eqs. (34) and (35) only have an effect on ${}^1\mu_{k,i}$)

$$ {}^1m_{k,i}^{\text{new}} = {}^1m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{\underline{f}^k}{\sum_{j=1}^{R} \underline{f}_r^j} \cdot \frac{x_i - {}^1m_{k,i}}{(\sigma_{k,i})^2}, \tag{34} $$

$$ \sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{\underline{f}^k}{\sum_{j=1}^{R} \underline{f}_r^j} \cdot \frac{(x_i - {}^1m_{k,i})^2}{(\sigma_{k,i})^3}. \tag{35} $$

$k > R$:
$x_i < {}^1m_{k,i}$ (Eqs. (36) and (37) only have an effect on ${}^1\mu_{k,i}$)

$$ {}^1m_{k,i}^{\text{new}} = {}^1m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{\bar{f}^k}{m_{j=R+1}^{M} \bar{f}_r^j} \cdot \frac{x_i - {}^1m_{k,i}}{(\sigma_{k,i})^2}, \tag{36} $$

$$ \sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{\bar{f}^k}{\sum_{j=R+1}^{M} \bar{f}_r^j} \cdot \frac{(x_i - {}^1m_{k,i})^2}{(\sigma_{k,i})^3}. \tag{37} $$

$x_i > {}^2m_{k,i}$ (Eqs. (38) and (39) only have an effect on ${}^2\mu_{k,i}$)

$$ {}^2m_{k,i}^{\text{new}} = {}^2m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{\bar{f}^k}{\sum_{j=R+1}^{M} \bar{f}_r^j} \cdot \frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2}, \tag{38} $$

$$ \sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{\bar{f}^k}{\sum_{j=R+1}^{M} \bar{f}_r^j} \cdot \frac{(x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3}. \tag{39} $$

The second architecture IT2FNN (Fig. 7) is an adaptive six-layer NN. The nodes of the first layer fuzzify the inputs, each node is a type-1 fuzzy neuron. Nodes from the second layer are *T*-norm type-1 fuzzy neurons whose outputs are the lower membership function and *S*-norm type-1 fuzzy neurons whose outputs are the upper membership functions. Layer three contains non-adaptive nodes that represent *lower–upper* firing set of the antecedents of the rules. In layer four each node evaluates the consequents rules. In layer five left–right points are evaluated using the Karnik–Mendel type-reduction algorithm [2]. Layer six defuzzifies the output of the system.

The forward-propagation procedure is described as follows:

*Layer 0*: Inputs

$$ x = (x_1, \ldots, x_i, \ldots, x_n)^t. $$

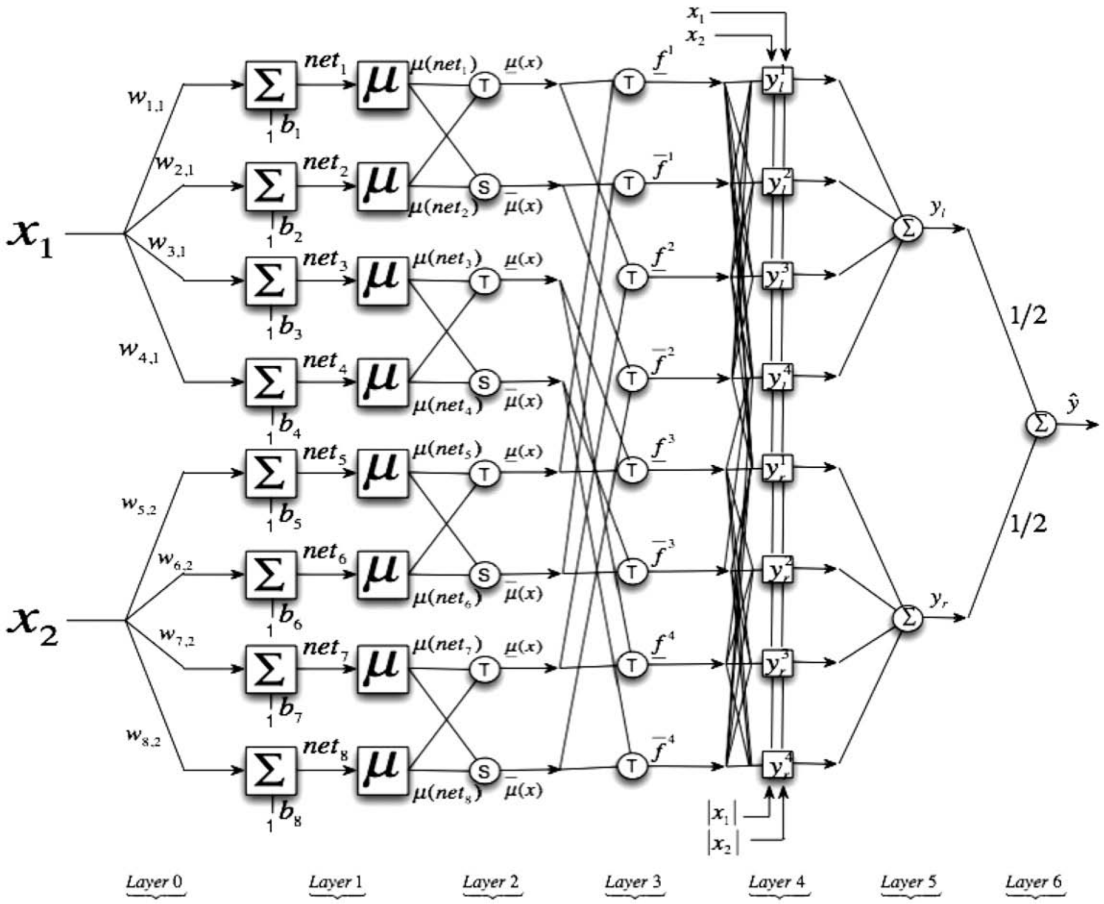


**Fig. 7.** Interval type-2 fuzzy neural network.

*Layer 1*: Every node $\ell$ in this layer is a square (Fig. 7) with a node function.

for $k = 1$ to $M$

for $i = 1$ to $n$

$$^{1}net_{k,i} = {}^{1}w_{k,i}x_i + {}^{1}b_{k,i};\ {}^{1}\mu_{k,i}(x_i) = \mu({}^{1}net_{k,i}),$$
$$^{2}net_{k,i} = {}^{2}w_{k,i}x_i + {}^{2}b_{k,i};\ {}^{2}\mu_{k,i}(x_i) = \mu({}^{2}net_{k,i}),$$

end

end

where $\mu$ is the transfer function (which can be Gaussian, gbell or logistic). For example Gaussian with uncertain mean (igaussmsttype2) is defined as

$$^{1}\mu_{k,i}(x_i) = e^{-{}^{1}net_{k,i}^{2}} = e^{-1/2\left(\frac{x_i - {}^{1}m_{k,i}}{\sigma_{k,i}}\right)^2},$$
$$^{2}\mu_{k,i}(x_i) = e^{-{}^{2}net_{k,i}^{2}} = e^{-1/2\left(\frac{x_i - {}^{2}m_{k,i}}{\sigma_{k,i}}\right)^2}$$

and the transfer function gbell with uncertain mean (igbellmsttype2) is defined as

$$^{1}\mu_{k,i}(x_i) = \frac{1}{1 + ({}^{1}net_{k,i}^{2})^{b_{k,i}}} = \frac{1}{1 + \left[\left(\frac{x_i - {}^{1}m_{k,i}}{a_{k,i}}\right)^2\right]^{b_{k,i}}},$$
$$^{2}\mu_{k,i}(x_i) = \frac{1}{1 + ({}^{2}net_{k,i}^{2})^{b_{k,i}}} = \frac{1}{1 + \left[\left(\frac{x_i - {}^{2}m_{k,i}}{a_{k,i}}\right)^2\right]^{b_{k,i}}},$$
$$k = 1, 2, \ldots, M; \quad i = 1, 2, \ldots, n.$$

*Layer 2*: Every node $\ell$ in this layer is a circle labeled with $T$-norm and $S$-norm alternated:

$$\underline{\mu}_{k,i}(x_i) = {}^1\mu_{k,i}(x_i)\cdot {}^2\mu_{k,i}(x_1);\ \bar{\mu}_{k,i}(x_i) = {}^1\mu_{k,i}(x_i) + {}^2\mu_{k,i}(x_i) - \underline{\mu}_{k,i}(x_i),$$
$$k = 1,2,\ldots,M;\quad i = 1,2,\ldots,n.$$

*Layer 3*: Rules

$$\underline{f}^k = \overset{n}{\underset{i=1}{\tilde{*}}}(\underline{\mu}_{k,i});\quad \bar{f}^k = \overset{n}{\underset{i=1}{\tilde{*}}}(\bar{\mu}_{k,i}).$$

*Layer 4*: Consequent left–right firing points

$$y_l^k = \sum_{i=1}^{n} c_{k,i}x_i + c_{k,0} - \sum_{i=1}^{n} s_{k,i}|x_i| - s_{k,0},$$
$$y_r^k = \sum_{i=1}^{n} c_{k,i}x_i + c_{k,0} + \sum_{i=1}^{n} s_{k,i}|x_i| + s_{k,0}.$$

*Layer 5*: Left and right firing points (type-reduction)

$$\hat{y}_l = \hat{y}_l(\bar{f}^1,\ldots,\bar{f}^L,\underline{f}^{L+1},\ldots,\underline{f}^M,y_l^1,\ldots,y_l^M) = \frac{\sum_{k=1}^{M} f_l^k\cdot y_l^k}{\sum_{k=1}^{M} f_l^k} = \frac{\sum_{k=1}^{L}\bar{f}^k\cdot y_l^k + \sum_{k=L+1}^{M}\underline{f}^k\cdot y_l^k}{\sum_{k=1}^{L}\bar{f}^k + \sum_{k=L+1}^{M}\underline{f}^k},$$
$$\hat{y}_r = \hat{y}_r(\underline{f}^1,\ldots,\underline{f}^R,\bar{f}^{R+1},\ldots,\bar{f}^M,y_r^1,\ldots,y_r^M) = \frac{\sum_{k=1}^{M} f_r^k\cdot y_r^k}{\sum_{k=1}^{M} f_r^k} = \frac{\sum_{k=1}^{R}\underline{f}^k\cdot y_r^k + \sum_{k=R+1}^{M}\bar{f}^k\cdot y_r^k}{\sum_{k=1}^{R}\underline{f}^k + \sum_{k=R+1}^{M}\bar{f}^k}.$$

*Layer 6*: Defuzzification

$$\hat{y} = \frac{\hat{y}_l + \hat{y}_r}{2}.$$

Equations from Eqs. (16)–(19) update the parameters of the consequents and (20)–(23) update the parameters of the antecedents of the rules.

For example: if $\mu_{k,i}(x_i) = \{\underline{\mu}_{k,i}(x_i),\bar{\mu}_{k,i}(x_i)\} = igaussmsttype2(x_i,[\sigma_{k,i},{}^1m_{k,i},{}^2m_{k,i}])$; where $k = 1,2,\ldots,M$; $i = 1,2,\ldots,n$; then equations from (41)–(52) update the parameters of the antecedents.

$k \leqslant L$:

$$ {}^1m_{k,i}^{\text{new}} = {}^1m_{k,i}^{\text{old}} + \eta\cdot 1/2\cdot e_p\cdot\frac{y_l^k-\hat{y}_l}{D_l}\cdot\left[(\bar{f}^k - {}^2\mu_{k,i}(x_i)\cdot\overset{n}{\underset{l=1,l\neq i}{\tilde{*}}}(\bar{\mu}_{k,l}))\cdot\frac{x_i - {}^1m_{k,i}}{(\sigma_{k,i})^2}\right], \tag{40}$$

$$ {}^2m_{k,i}^{\text{new}} = {}^2m_{k,i}^{\text{old}} + \eta\cdot 1/2\cdot e_p\cdot\frac{y_l^k-\hat{y}_l}{D_l}\cdot\left[(\bar{f}^k - {}^1\mu_{k,i}(x_i)\cdot\overset{n}{\underset{l=1,l\neq i}{\tilde{*}}}(\bar{\mu}_{k,l}))\cdot\frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2}\right], \tag{41}$$

$$\begin{aligned}\sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta\cdot 1/2\cdot e_p\cdot\frac{y_l^k-\hat{y}_l}{D_l}\cdot&\left[\left(\bar{f}^k - {}^2\mu_{k,i}(x_i)\cdot\overset{n}{\underset{l=1,l\neq i}{\tilde{*}}}(\bar{\mu}_{k,l})\right)\cdot\frac{(x_i - {}^1m_{k,i})^2}{(\sigma_{k,i})^3}\right.\\ &\left.+\left(\bar{f}^k - {}^1\mu_{k,i}(x_i)\cdot\overset{n}{\underset{l=1,l\neq i}{\tilde{*}}}(\bar{\mu}_{k,l})\right)\cdot\frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2}\right].\end{aligned} \tag{42}$$

$k > L$:

$$ {}^1m_{k,i}^{\text{new}} = {}^1m_{k,i}^{\text{old}} + \eta\cdot 1/2\cdot e_p\cdot(y_l^k-\hat{y}_l)\cdot\frac{\underline{f}^k}{D_l}\cdot\frac{x_i - {}^1m_{k,i}}{(\sigma_{k,i})^2}, \tag{43}$$

$$ {}^2m_{k,i}^{\text{new}} = {}^2m_{k,i}^{\text{old}} + \eta\cdot 1/2\cdot e_p\cdot(y_l^k-\hat{y}_l)\cdot\frac{\underline{f}^k}{D_l}\cdot\frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2}, \tag{44}$$

$$ {}^1\sigma_{k,i}^{\text{new}} = {}^1\sigma_{k,i}^{\text{old}} + \eta\cdot 1/2\cdot e_p\cdot(y_l^k-\hat{y}_l)\cdot\frac{\underline{f}^k}{D_l}\cdot\left[\frac{(x_i - {}^1m_{k,i})^2 + (x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3}\right]. \tag{45}$$

$k \leqslant R$:

$$ {}^1m_{k,i}^{\text{new}} = {}^1m_{k,i}^{\text{old}} + \eta\cdot 1/2\cdot e_p\cdot(y_r^k-\hat{y}_r)\cdot\frac{\underline{f}^k}{D_r}\cdot\frac{x_i - {}^1m_{k,i}}{(\sigma_{k,i})^2}, \tag{46}$$

$$ {}^2m_{k,i}^{\text{new}} = {}^2m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{f^k}{\overline{D}_r} \cdot \frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2}, \tag{47} $$

$$ {}^1\sigma_{k,i}^{\text{new}} = {}^1\sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot (y_r^k - \hat{y}_r) \cdot \frac{f^k}{\overline{D}_r} \cdot \left[ \frac{(x_i - {}^1m_{k,i})^2 + (x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3} \right]. \tag{48} $$

$k > R$:

$$ {}^1m_{k,i}^{\text{new}} = {}^1m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_r^k - \hat{y}_r}{D_r} \cdot \left[ \left( \bar{f}^k - {}^2\mu_{k,i}(x_i) \cdot \mathop{\tilde{*}}_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right) \cdot \frac{x_i - {}^1m_{k,i}}{(\sigma_{k,i})^2} \right], \tag{49} $$

$$ {}^2m_{k,i}^{\text{new}} = {}^2m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_r^k - \hat{y}_r}{D_r} \cdot \left[ \left( \bar{f}^k - {}^1\mu_{k,i}(x_i) \cdot \mathop{\tilde{*}}_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right) \cdot \frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2} \right], \tag{50} $$

$$ \begin{aligned} \sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_r^k - \hat{y}_r}{D_r} \cdot & \left[ \left( \bar{f}^k - {}^2\mu_{k,i}(x_i) \cdot \mathop{\tilde{*}}_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right) \cdot \frac{(x_i - {}^1m_{k,i})^2}{(\sigma_{k,i})^3} \right. \\ & \left. + \left( \bar{f}^k - {}^1\mu_{k,i}(x_i) \cdot \mathop{\tilde{*}}_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right) \cdot \frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2} \right]. \end{aligned} \tag{51} $$

The third architecture IT2FNN implements an TSK-IT2FIS and has a seven layered architecture as shown in Fig. 8. The first and second hidden layers are for fuzzifying input variables, and $T$-norm operators are deployed in the third hidden layer to compute the rule antecedent part. The fourth hidden layer evaluates firing left-most and right-most points, $f^k(x) \in [\underline{f}^k(x), \bar{f}^k(x)]$, the rule strengths followed by the fifth hidden layer where the consequent for each rule are determined by $\tilde{y}_k^j \in [{}_l y_k^j, {}_r y_k^j]$ (see Figs. 5 and 8). The sixth layer computes its two end-points, ${}_l y^j$ and ${}_r y^j$, and in the seventh layer, we defuzzify the $j$th output using the average of ${}_l y^j$ and ${}_r y^j$.

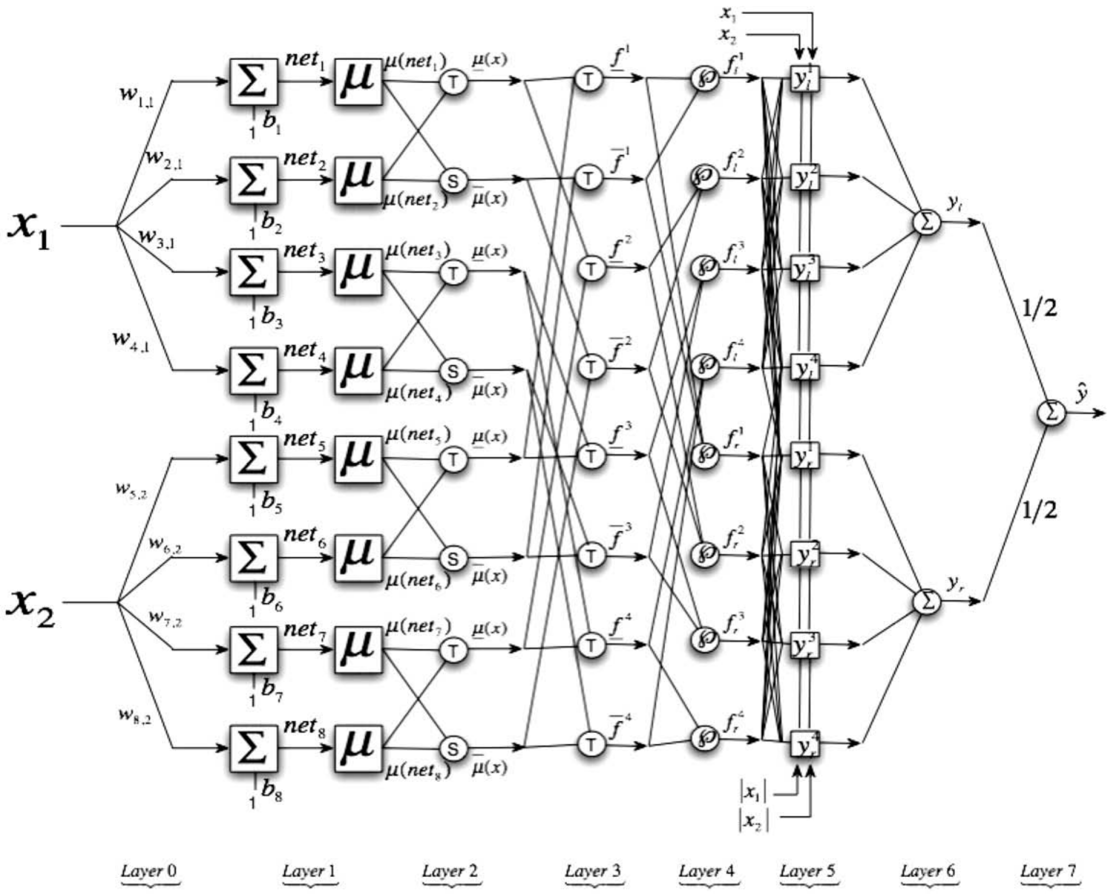


**Fig. 8.** Interval type-2 fuzzy neural network.

The forward-propagation procedure is described as follows:

*Layer* 0: Inputs

$$x = (x_1, \ldots, x_i, \ldots, x_n)^t$$

*Layer 1*: Every node $\ell$ in this layer is a square (Fig. 6) with a node function.

for $k = 1$ to $M$

for $i = 1$ to $n$

$$^1net_{k,i} = {}^1w_{k,i}x_i + {}^1b_{k,i}; \quad {}^1\mu_{k,i}(x_i) = \mu({}^1net_{k,i}),$$
$$^2net_{k,i} = {}^2w_{k,i}x_i + {}^2b_{k,i}; \quad {}^2\mu_{k,i}(x_i) = \mu({}^2net_{k,i}),$$

end

end

where $\mu$ is this transfer function (which can be Gaussian, gbell or logistic). For example Gaussian with uncertain mean:

$$^1\mu_{k,i}(x_i) = e^{-{}^1net_{k,i}^2} = e^{-1/2\left(\frac{x_i - {}^1m_{k,i}}{\sigma_{k,i}}\right)^2},$$
$$^2\mu_{k,i}(x_i) = e^{-{}^2net_{k,i}^2} = e^{-1/2\left(\frac{x_i - {}^2m_{k,i}}{\sigma_{k,i}}\right)^2},$$

or gbell transfer function:

$$^1\mu_{k,i}(x_i) = \frac{1}{1 + ({}^1net_{k,i}^2)^{b_{k,i}}} = \frac{1}{1 + \left[\left(\frac{x_i - {}^1m_{k,i}}{a_{k,i}}\right)^2\right]^{b_{k,i}}},$$
$$^2\mu_{k,i}(x_i) = \frac{1}{1 + ({}^2net_{k,i}^2)^{b_{k,i}}} = \frac{1}{1 + \left[\left(\frac{x_i - {}^2m_{k,i}}{a_{k,i}}\right)^2\right]^{b_{k,i}}},$$
$$k = 1, 2, \ldots, M; \quad i = 1, 2, \ldots, n.$$

*Layer 2*: Every node $\ell$ in this layer is a circle labeled with $T$-norm and $S$-norm alternated:

$$\underline{\mu}_{k,i}(x_i) = {}^1\mu_{k,i}(x_i) \cdot {}^2\mu_{k,i}(x_i); \bar{\mu}_{k,i}(x_i) = {}^1\mu_{k,i}(x_i) + {}^2\mu_{k,i}(x_i) - \underline{\mu}_{k,i}(x_i),$$
$$k = 1, 2, \ldots, M; \quad i = 1, 2, \ldots, n.$$

*Layer 3*: Every node $\ell$ in this layer is a circle labeled $T$, which multiplies the incoming signals and sends the product out. Each output node represents the lower ($\underline{f}^k$) and upper ($\bar{f}^k$) firing strength of a rule:

$$\underline{\underline{f}^k = \overset{n}{\underset{i=1}{\tilde{*}}}(\underline{\mu}_{k,i});} \quad \underline{\bar{f}^k = \overset{n}{\underset{i=1}{\tilde{*}}}(\bar{\mu}_{k,i}).}$$

*Layer 4*: Every node $\ell$ in this layer is a circle labeled $\wp$ which evaluates the left-most and right-most firing points denoted by

$$f_l^k = \frac{\bar{\omega}_l^k \bar{f}^k + \underline{\omega}_l^k \underline{f}^k}{\bar{\omega}_l^k + \underline{\omega}_l^k}; \quad f_r^k = \frac{\underline{\omega}_r^k \underline{f}^k + \bar{\omega}_r^k \bar{f}^k}{\underline{\omega}_r^k + \bar{\omega}_r^k},$$

where $\omega$ are adjustable weights.

*Layer 5*: Every node $\ell$ in this layer is a square labeled $y_l$ and $y_r$, which computes $y_l^k, y_r^k$:

$$y_l^k = \sum_{i=1}^n c_{k,i}x_i + c_{k,0} - \sum_{i=1}^n s_{k,i}|x_i| - s_{k,0},$$
$$y_r^k = \sum_{i=1}^n c_{k,i}x_i + c_{k,0} + \sum_{i=1}^n s_{k,i}|x_i| + s_{k,0}.$$

*Layer 6*: The two nodes in this layer are circles labeled with "$\sum$" that evaluates the two end-points, $y_l$ and $y_r$

$$\hat{y}_l = \hat{y}_l(\bar{f}^1, \ldots, \bar{f}^L, \underline{f}^{L+1}, \ldots, \underline{f}^M, y_l^1, \ldots, y_l^M) = \frac{\sum_{k=1}^M f_l^k \cdot y_l^k}{\sum_{k=1}^M f_l^k},$$
$$\hat{y}_r = \hat{y}_r(\underline{f}^1, \ldots, \underline{f}^R, \bar{f}^{R+1}, \ldots, \bar{f}^M, y_r^1, \ldots, y_r^M) = \frac{\sum_{k=1}^M f_r^k \cdot y_r^k}{\sum_{k=1}^M f_r^k}.$$

*Layer 7*: The single node in this layer is a circle labeled "$\sum$" that computes the output.

$$\hat{y} = \frac{\hat{y}_l + \hat{y}_r}{2}.$$

Equations from (16)–(19) update the parameters of the consequents of the rules and (52)–(55) update the left–right-most firing set points for type-reduction

$$^{\text{new}}\underline{\omega}_l^k = {}^{\text{old}}\underline{\omega}_l^k + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_l^k - \hat{y}_l}{\sum_{j=1}^{M} f_l^j} \cdot \frac{\underline{f}^k - f_l^k}{\bar{\omega}_l^k + \underline{\omega}_l^k}, \tag{52}$$

$$^{\text{new}}\bar{\omega}_l^k = {}^{\text{old}}\bar{\omega}_l^k + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_l^k - \hat{y}_l}{\sum_{j=1}^{M} f_l^j} \cdot \frac{\bar{f}^k - f_l^k}{\bar{\omega}_l^k + \underline{\omega}_l^k}, \tag{53}$$

$$^{\text{new}}\underline{\omega}_r^k = {}^{\text{old}}\underline{\omega}_r^k + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_r^k - \hat{y}_r}{\sum_{j=1}^{M} f_r^j} \cdot \frac{\underline{f}^k - f_r^k}{\underline{\omega}_r^k + \bar{\omega}_r^k}, \tag{54}$$

$$^{\text{new}}\bar{\omega}_r^k = {}^{\text{old}}\bar{\omega}_r^k + \eta \cdot 1/2 \cdot e_p \cdot \frac{y_r^k - \hat{y}_r}{\sum_{j=1}^{M} f_r^j} \cdot \frac{\bar{f}^k - f_r^k}{\underline{\omega}_r^k + \bar{\omega}_r^k}, \tag{55}$$

$$\frac{\partial f_l^k}{\partial \xi_{k,i}} = \frac{\bar{\omega}_l^k \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \cdot \frac{\partial \bar{\mu}_{k,i}}{\partial \xi_{k,i}} + \underline{\omega}_l^k \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\underline{\mu}_{k,l}) \cdot \frac{\partial \underline{\mu}_{k,i}}{\partial \xi_{k,i}}}{\bar{\omega}_l^k + \underline{\omega}_l^k}, \tag{56}$$

$$\frac{\partial f_r^k}{\partial \xi_{k,i}} = \frac{\underline{\omega}_r^k \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\underline{\mu}_{k,l}) \cdot \frac{\partial \underline{\mu}_{k,i}}{\partial \xi_{k,i}} + \bar{\omega}_r^k \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \cdot \frac{\partial \bar{\mu}_{k,i}}{\partial \xi_{k,i}}}{\underline{\omega}_r^k + \bar{\omega}_r^k}, \tag{57}$$

$$^{\text{new}}\xi_{k,i} = {}^{\text{old}}\xi_{k,i} + \eta \cdot 1/2 \cdot e_p \left[ \frac{y_l^k - \hat{y}_l}{\sum_{j=1}^{M} f_l^j} \cdot \frac{\partial f_l^k}{\partial \xi_{k,i}} + \frac{y_r^k - \hat{y}_r}{\sum_{j=1}^{M} f_r^j} \cdot \frac{\partial f_r^k}{\partial \xi_{k,i}} \right] \tag{58}$$

for example: if $\mu_{k,i}(x_i) = \{\underline{\mu}_{k,i}(x_i), \bar{\mu}_{k,i}(x_i)\} = igaussmsttype2(x_i, [\sigma_{k,i}, {}^1m_{k,i}, {}^2m_{k,i}])$; where $k = 1,2,\ldots,M$; $i = 1,2,\ldots,n$ then all derivatives related to antecedents of rules are given by

$$\frac{\partial f_l^k}{\partial {}^1m_{k,i}} = \frac{\bar{\omega}_l^k \cdot \left[ \bar{f}^k - {}^2\mu_{k,i} \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right] + \underline{\omega}_l^k \cdot \underline{f}^k}{\bar{\omega}_l^k + \underline{\omega}_l^k} \cdot \frac{x_i - {}^1m_{k,i}}{({}^1\sigma_{k,i})^2}, \tag{59}$$

$$\frac{\partial f_l^k}{\partial {}^2m_{k,i}} = \frac{\bar{\omega}_l^k \cdot \left[ \bar{f}^k - {}^1\mu_{k,i} \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right] + \underline{\omega}_l^k \cdot \underline{f}^k}{\bar{\omega}_l^k + \underline{\omega}_l^k} \cdot \frac{x_i - {}^2m_{k,i}}{({}^2\sigma_{k,i})^2}, \tag{60}$$

$$\begin{aligned} \frac{\partial f_l^k}{\partial \sigma_{k,i}} = {} & \frac{\bar{\omega}_l^k \cdot \left[ (\bar{f}^k - {}^2\mu_{k,i} \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l})) \cdot \frac{(x_i - {}^1m_{k,i})^2}{(\sigma_{k,i})^3} \right]}{\bar{\omega}_l^k + \underline{\omega}_l^k} + \frac{\bar{\omega}_l^k \cdot \left[ (\bar{f}^k - {}^1\mu_{k,i} \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l})) \cdot \frac{(x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3} \right]}{\bar{\omega}_l^k + \underline{\omega}_l^k} \\ & + \frac{\underline{\omega}_l^k \cdot \underline{f}^k \cdot \left[ \frac{(x_i - {}^1m_{k,i})^2 + (x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3} \right]}{\bar{\omega}_l^k + \underline{\omega}_l^k}, \end{aligned} \tag{61}$$

$$\frac{\partial f_r^k}{\partial {}^1m_{k,i}} = \frac{\underline{\omega}_r^k \cdot \underline{f}^k + \bar{\omega}_r^k \cdot \left[ \bar{f}^k - {}^2\mu_{k,i} \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right]}{\underline{\omega}_r^k + \bar{\omega}_r^k} \cdot \frac{x_i - {}^1m_{k,i}}{(\sigma_{k,i})^2}, \tag{62}$$

$$\frac{\partial f_r^k}{\partial {}^2m_{k,i}} = \frac{\underline{\omega}_r^k \cdot \underline{f}^k + \bar{\omega}_r^k \cdot \left[ \bar{f}^k - {}^1\mu_{k,i} \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right]}{\underline{\omega}_r^k + \bar{\omega}_r^k} \cdot \frac{x_i - {}^2m_{k,i}}{(\sigma_{k,i})^2}, \tag{63}$$

$$\begin{aligned} \frac{\partial f_r^k}{\partial \sigma_{k,i}} = {} & \frac{\underline{\omega}_r^k \cdot \underline{f}^k \cdot \left[ \frac{(x_i - {}^1m_{k,i})^2 + (x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3} \right]}{\underline{\omega}_r^k + \bar{\omega}_r^k} + \frac{\bar{\omega}_r^k \cdot \left[ \left( \bar{f}^k - {}^2\mu_{k,i} \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right) \cdot \frac{(x_i - {}^1m_{k,i})^2}{(\sigma_{k,i})^3} \right]}{\underline{\omega}_r^k + \bar{\omega}_r^k} \\ & + \frac{\bar{\omega}_r^k \cdot \left[ \left( \bar{f}^k - {}^1\mu_{k,i} \cdot \mathop{\tilde{\ast}}\limits_{l=1,l\neq i}^{n} (\bar{\mu}_{k,l}) \right) \cdot \frac{(x_i - {}^2m_{k,i})^2}{(\sigma_{k,i})^3} \right]}{\underline{\omega}_r^k + \bar{\omega}_r^k}, \end{aligned} \tag{64}$$

$${}^1m_{k,i}^{\text{new}} = {}^1m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \left[ \frac{y_l^k - \hat{y}_l}{\sum_{j=1}^{M} f_l^j} \cdot \frac{\partial f_l^k}{\partial {}^1m_{k,i}} + \frac{y_r^k - \hat{y}_r}{\sum_{j=1}^{M} f_r^j} \cdot \frac{\partial f_r^k}{\partial {}^1m_{k,i}} \right], \tag{65}$$

$$ {}^{2}m_{k,i}^{\text{new}} = {}^{2}m_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \left[ \frac{y_l^k - \hat{y}_l}{\sum_{j=1}^{M} f_l^j} \cdot \frac{\partial f_l^k}{\partial^2 m_{k,i}} + \frac{y_r^k - \hat{y}_r}{\sum_{j=1}^{M} f_r^j} \cdot \frac{\partial f_r^k}{\partial^2 m_{k,i}} \right], \tag{66} $$

$$ \sigma_{k,i}^{\text{new}} = \sigma_{k,i}^{\text{old}} + \eta \cdot 1/2 \cdot e_p \left[ \frac{y_l^k - \hat{y}_l}{\sum_{j=1}^{M} f_l^j} \cdot \frac{\partial f_l^k}{\partial \sigma_{k,i}} + \frac{y_r^k - \hat{y}_r}{\sum_{j=1}^{M} f_r^j} \cdot \frac{\partial f_r^k}{\partial \sigma_{k,i}} \right]. \tag{67} $$

## 4. Simulation results

We present results from simulations of three different IT2FNN architectures for the TSK model for the on-line identification in a control system, and Mackey–Glass chaotic series prediction. First column in Table 1 show the indexes of hybrid models, first index represents the architecture and second index the transference function type where 1 is for Gaussian symmetric (igaussmtype2) with uncertain mean, and 2 is for an asymmetric transference function (igausssttype2) with uncertain mean and standard deviation.

### 4.1. On-line identification in control systems

In this paper, we compare our IT2FNN with a simulation example given in [1,18], where a 1-20-10-1 backpropagation MLP is employed to identify a nonlinear component in a control system. The plant under consideration is governed by the following difference equation:

$$ y(k+1) = 0.3y(k) + 0.6^{*}y(k-1) + f(u(k)), \tag{68} $$

where $y(k)$ and $u(k)$ are the output and input, respectively, at time step $k$. The unknown function $f(\cdot)$ has the form

$$ f(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u). \tag{69} $$

In order to identify the plant, a series-parallel model governed by the difference equation

$$ \hat{y}(k+1) = 0.3\hat{y}(k) + 0.6 * \hat{y}(k-1) + F(u(k)) \tag{70} $$

was used, where $F(\cdot)$ is the function implemented by the IT2FNN and its parameters are updated at each time step.

The number of membership functions assigned to each input of the IT2FNN was arbitrarily set to 5, so the rule number is 5. After 50 epochs (Table 1) we obtained, using IT2FNN (which performed better than ANFIS), a RMSE = 0.0055 (training).

In Table 1 and Fig. 9, it is shown that the proposed architectures clearly identify the control function and the best architecture is the third one (Models 31 and 32) with asymmetric transference functions.

### 4.2. Predicting chaotic time series

The time series used for the simulation was generated by the chaotic Mackey–Glass differential delay equation [17] defined below:

$$ \dot{x}(t) = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t). \tag{71} $$

For $\tau > 17$ is known to exhibit chaos.

The prediction of future values of these time series is a benchmark problem, which has been considered by a number of connectionist researchers [18,37].

The goal is to use known values of the time series up to the point $x = t$ to predict the value at some point in the future $x = t + P$. The standard method for this type of prediction is to create a mapping from $D$ points of the time series spaced $\Delta$ units apart, that is $(x(t-(D-1)\Delta), \ldots, x(t-\Delta), x(t))$, to a predicted future value $x(t+P)$. To allow comparison with earlier

**Table 1**
Training RMSE After 50 epochs.

| Hybrid model | MF | RMSE |
|---|---|---|
| 11 | igaussmtype2[a*] | 0.0061 |
| 12 | igausssttype2[a] | 0.0064 |
| 21 | igaussmtype2 | 0.0063 |
| 22 | igausssttype2 | 0.0057 |
| 31 | igaussmtype2 | 0.0056 |
| 32 | igausssttype2 | 0.0055 |
| ANFIS | gaussmf[b] | 0.0070 |

[a] Interval type-2 membership function.
[b] Type-1 membership function.

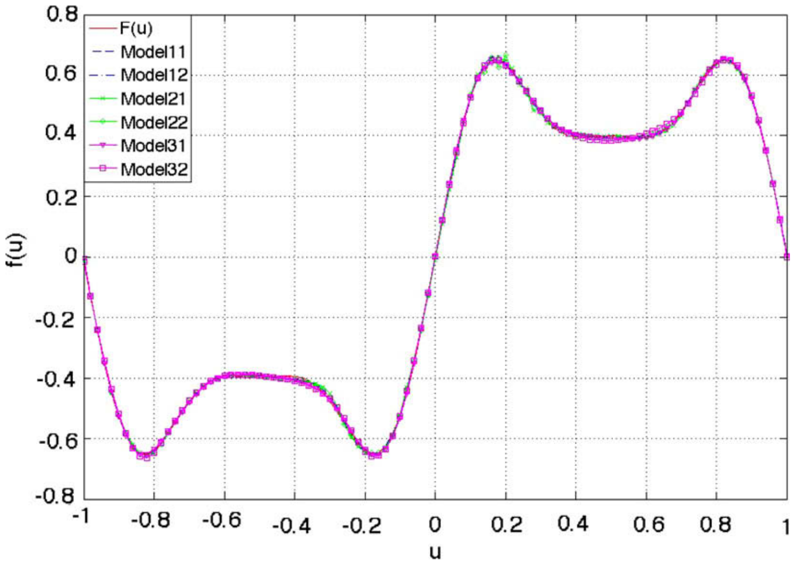


**Fig. 9.** Off-line learning with five rules.

work [18], the values $D = 4$ and $\Delta = P = 6$ were used. All other simulation setting in this example were purposely arranged to be as close as possible to those reported in [19].

To obtain the time series value at each integer point, we applied the fourth-order Runge–Kutta method to find the numerical solution to the equation. The time step used in the method is 0.1, initial condition $x(0) = 1.2$, $\tau = 13, 17, 30, 100$ and $x(t)$ is thus derived for $0 \leqslant t \leqslant 1200$. From the Mackey–Glass time series $x(t)$, we extracted 1000 input–output data pairs of the following format:

$$[x(t-24), x(t-18), x(t-12), x(t-6); x(t)],$$

where $t = 124$ to 1123. The first 500 pairs (training data set) were used for training the IT2FNN while the remaining 500 pairs (checking data set) were used for validating the identified model (see Figs. 10–14).

### 4.3. Experiment 1: Mackey–Glass time series prediction with τ = 13, 17, 30, 100 noiseless, using IT2FNN

IT2FNN uses gradient descent with adaptive learning rate backpropagation to identify TSK-IT2FIS parameters. The number of membership functions assigned to each input of the IT2FNN was arbitrarily set to 2, so the number of rules is 16. Cha-

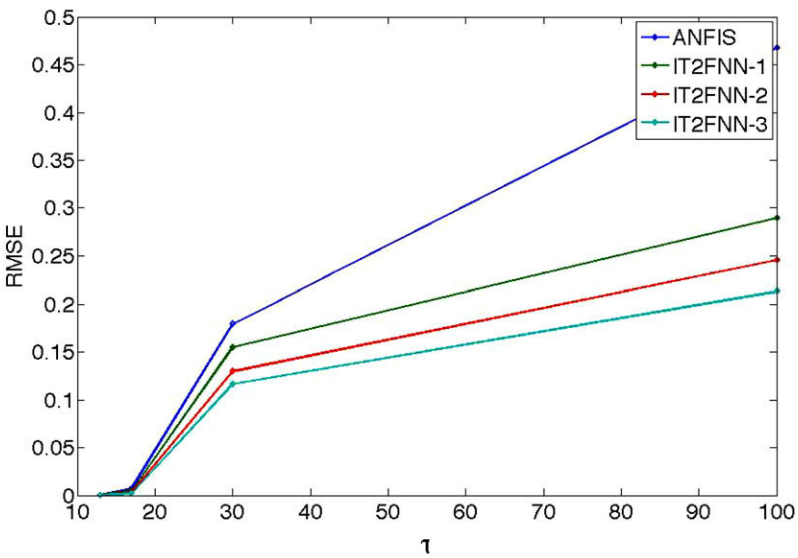


**Fig. 10.** Mackey–Glass time series prediction with $\tau$ noiseless, using IT2FNN.

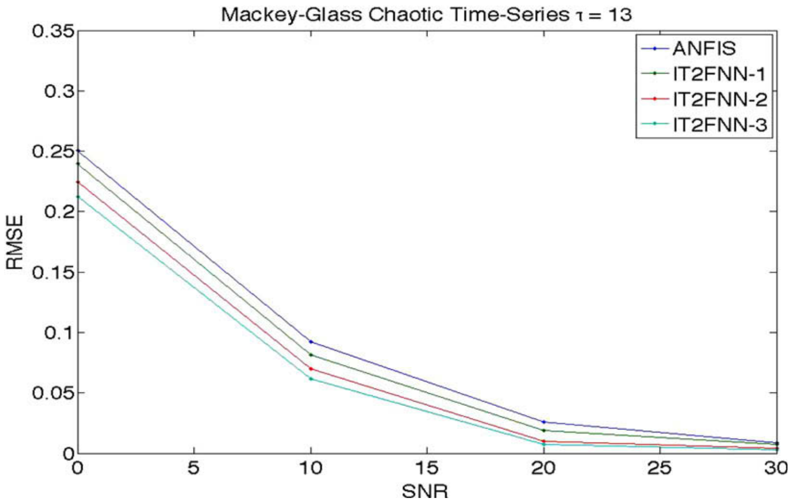


**Fig. 11.** Mackey–Glass time series prediction for $\tau = 13$ corrupted with noise using IT2FNN.

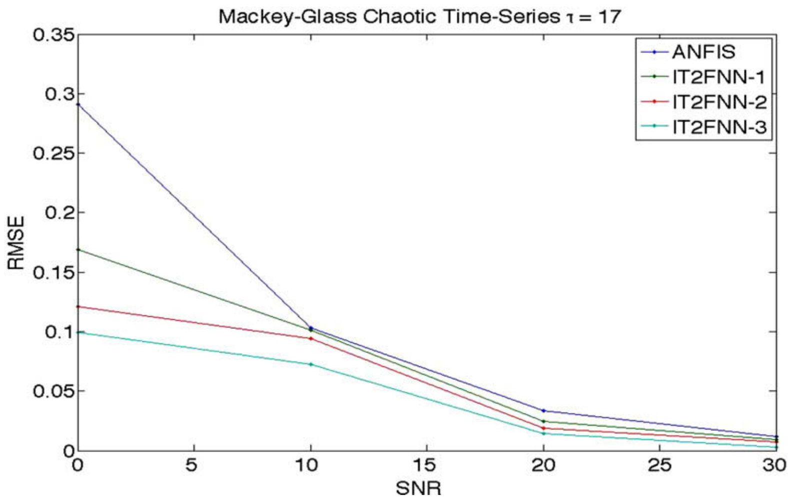


**Fig. 12.** Mackey–Glass time series prediction for $\tau = 17$ corrupted with noise using IT2FNN.

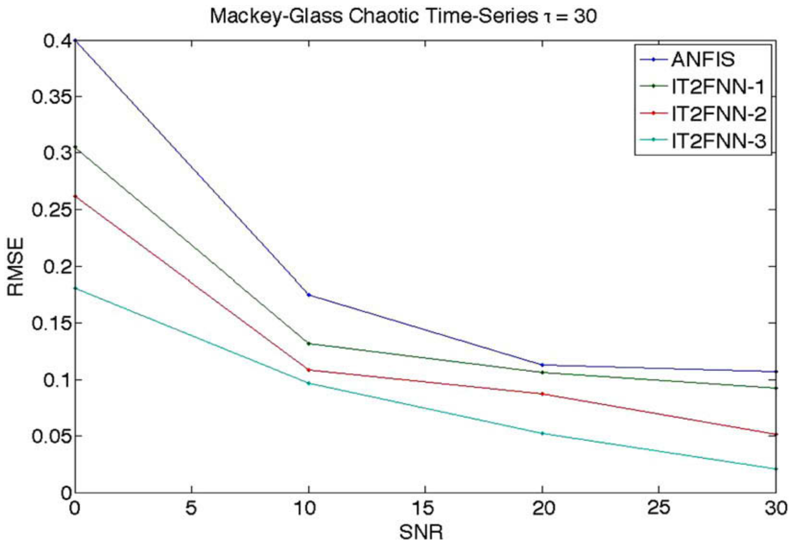


**Fig. 13.** Mackey–Glass time series prediction for $\tau = 30$ corrupted with noise using IT2FNN.

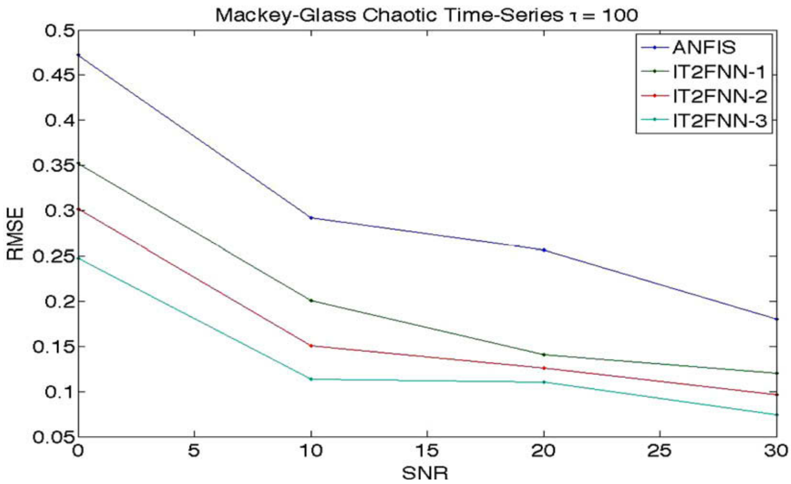


**Fig. 14.** Mackey–Glass time series prediction for $\tau = 100$ corrupted with noise using IT2FNN.

otic degree ($\tau$) from Mackey–Glass series is taken as an uncertainty source and variated to test performance of IT2FNN and ANFIS.

Table 2 shows IT2FNN architectures perform better than ANFIS when chaotic degree is increased (and thus uncertainty degree). Architectures with IT2FN's (IT2FNN-2, IT2FNN-3) performed better than those based on simple nodes with membership function type-1 lower and upper (IT2FNN-1).

### *4.4. Experiment 2: Mackey–Glass time series prediction for $\tau = 13, 17, 30, 100$ corrupted with noise using IT2FNN*

IT2FNN uses gradient descent with adaptive learning rate backpropagation to identify TSK-IT2FIS parameters. Series with multiple chaotic degrees are corrupted with noise levels of 0 dB, 10 dB 20 dB and 30 dB of SNR (signal noise ratio) as a high source of uncertainty. Tables 3–6 show that when chaotic degree increases, and different levels of noise are added, a higher source of uncertainty is generated. Thus, IT2FNN perform better than ANFIS. This is due to the fact that IT2FNN handle uncertainty better and thus IT2FNN architectures increase learning skills.

**Table 2**
The noise-free Mackey–Glass chaotic time-series prediction ($\tau = 13, 17, 30, 100$).

| Model | RMSE ($\tau = 13$) | RMSE ($\tau = 17$) | RMSE ($\tau = 30$) | RMSE ($\tau = 100$) |
|---|---|---|---|---|
| ANFIS | 2.0196e−4 | 0.0070 | 0.1792 | 0.4678 |
| IT2FNN-1 | 2.0132e−4 | 0.0050 | 0.1548 | 0.2897 |
| IT2FNN-2 | 2.0112e−4 | 0.0035 | 0.1297 | 0.2456 |
| IT2FNN-3 | 2.0014e−4 | 0.0020 | 0.1165 | 0.2132 |

**Table 3**
Corrupted by uniformly-distributed stationary additive noise Mackey–Glass chaotic time-series prediction ($\tau = 13$).

| Model | SNR (0 dB) | SNR (10 dB) | SNR (20 dB) | SNR (30 dB) |
|---|---|---|---|---|
| ANFIS | 0.2506 | 0.0925 | 0.0257 | 0.0087 |
| IT2FNN-1 | 0.2397 | 0.0814 | 0.0187 | 0.0074 |
| IT2FNN-2 | 0.2247 | 0.0701 | 0.0097 | 0.0044 |
| IT2FNN-3 | 0.2125 | 0.0617 | 0.0076 | 0.0028 |

**Table 4**
Corrupted by uniformly-distributed stationary additive noise Mackey–Glass chaotic time series prediction ($\tau = 17$).

| Model | SNR (0 dB) | SNR (10 dB) | SNR (20 dB) | SNR (30 dB) |
|---|---|---|---|---|
| ANFIS | 0.2910 | 0.1031 | 0.0333 | 0.0115 |
| IT2FNN-1 | 0.1691 | 0.1013 | 0.0245 | 0.0092 |
| IT2FNN-2 | 0.1213 | 0.0941 | 0.0187 | 0.0071 |
| IT2FNN-3 | 0.0991 | 0.0722 | 0.0143 | 0.0028 |

**Table 5**
Corrupted by uniformly-distributed stationary additive noise Mackey–Glass chaotic time-series prediction ($\tau = 30$).

| Model | SNR (0 dB) | SNR (10 dB) | SNR (20 dB) | SNR (30 dB) |
|---|---|---|---|---|
| ANFIS | 0.4000 | 0.1746 | 0.1130 | 0.1070 |
| IT2FNN-1 | 0.3053 | 0.1314 | 0.1060 | 0.0924 |
| IT2FNN-2 | 0.2618 | 0.1087 | 0.0871 | 0.0513 |
| IT2FNN-3 | 0.1805 | 0.0969 | 0.0521 | 0.0207 |

**Table 6**
Corrupted by uniformly-distributed stationary additive noise Mackey–Glass chaotic time-series prediction ($\tau = 100$).

| Model | SNR (0 dB) | SNR (10 dB) | SNR (20 dB) | SNR (30 dB) |
|---|---|---|---|---|
| ANFIS | 0.4721 | 0.2916 | 0.2568 | 0.1799 |
| IT2FNN-1 | 0.3522 | 0.2009 | 0.1409 | 0.1199 |
| IT2FNN-2 | 0.3019 | 0.1503 | 0.1261 | 0.0968 |
| IT2FNN-3 | 0.2475 | 0.1137 | 0.1107 | 0.0745 |

## 5. Conclusions

This paper has presented an IT2FNN system and the corresponding backpropagation learning algorithm (gradient descent backpropagation and gradient descent with adaptive learning rate backpropagation). This IT2FNN is used to treat uncertainty associated with information or data that, due to the properties of interval type-2 fuzzy sets, it can represent and handle uncertain information effectively.

The IT2FIS has meaningful representations (interval type-2 fuzzy If–Then rules and interval type-2 fuzzy reasoning) derived from human expertise, but it has no adaptive capability (learning from examples) to take advantage of a desired input–output data set. On the other hand, NN represents a totally different paradigm with learning capability that adapts its parameters based on desired input–output pairs, but neither can accommodate *a priori* knowledge from humans experts, nor can we transform a network configuration and connection weights into a meaningful representation to account for structured knowledge.

Another important issue in training IT2FNN is how to preserve some intuitive features that make resulting interval type-2 fuzzy rules easy to interpret.

Throughout this paper, we have assumed that the structure of IT2FNN is fixed and that parameter identification is solved through hybrid learning rule. However, to make the whole approach more complete, structure identification (which is concerned with selection of an appropriate input-space partition style, number of membership functions on each input, and so on) is equally important to successful application of IT2FNN, especially for modeling problems with a large number of inputs. Effective partitioning of input space can decrease the number of rules and thus increase the speed in both learning and application phases.

Time series results show that intelligent hybrid methods can be derived as a generalization of autoregressive non-linear models in the context of time series. This derivation allows a practical specification for a general class of prognosis and identification time series models, where a set of input–output variables are part of the dynamics of time series knowledge base. This helps the application of the methodology to a series of diverse dynamics, with a very small number of causal variables to explain behavior.

## Acknowledgements

The authors would like to thank CONACYT and DGEST for the financial support given to this research project. The student (Juan R. Castro) is supported by a scholarship from UABC-CONACYT.

## References

[1] J.S.R. Jang, C.T. Sun, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice-Hall Inc., Upper Saddle River, NJ, 1996.
[2] J.M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, NJ, 2001.
[3] W. Pedrycz, Neurocomputations in relational systems, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-13 (3) (1991) 289–297.
[4] W. Pedrycz, A.F. Rocha, Fuzzy-set based model of neurons and knowledge-based networks, IEEE Transactions on Fuzzy Systems 1 (1993) 254–266.
[5] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[6] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man and Cybernetics 3 (1) (1973) 28–44.
[7] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning: Parts 1–3, Information Sciences 8 (1975) 199–249. 8, 301–357; 9, 43–80.
[8] L.A. Zadeh, Fuzzy logic, Computer 1 (4) (1988) 83–93.
[9] L.A. Zadeh, Knowledge representation in fuzzy logic, IEEE Transactions on Knowledge and Data Engineering 1 (1989) 89–100.
[10] L.A. Zadeh, Fuzzy logic = computing with words, IEEE Transactions on Fuzzy Systems 2 (1996) 103–111.

[11] N.N. Karnik, J.M. Mendel, An Introduction to Type-2 Fuzzy Logic Systems, Univ. of Southern Calif., Los Angeles, CA, 1998.
[12] Q. Liang, J. Mendel, Interval type-2 fuzzy logic systems: theory and design, IEEE Transactions on Fuzzy Systems 8 (2000) 535–550.
[13] M. Mizumoto, K. Tanaka, Some properties of fuzzy sets of type-2, Information and Control 31 (1976) 312–340.
[14] Juan R. Castro, Hybrid intelligent architecture development for time series forecasting, Masters Degree Thesis, Tijuana Institute of Technology, December 2005.
[15] (a) M. Sugeno, G.T. Kang, Structure identification of fuzzy model, Fuzzy Sets and Systems 28 (1988) 15–33;
(b) M. Sugeno, Industrial Applications of Fuzzy Control, Elsevier Science Pub. Co., 1985.
[16] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems, Man and Cybernetics 15 (1985) 116–132.
[17] M.C. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (1977) 287–289.
[18] S. Lapedes, R. Farber, Nonlinear signal processing using neural networks: prediction and system modeling, Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, 1987.
[19] R.S. Crowder, Predicting the Mackey–Glass time series with cascade-correlation learning, in: D. Touretzky, G. Hinton, T. Sejnowski (Eds.), Proceedings of the 1990 Connectionist Models Summer School, Carnegic Mellon University, 1990, pp. 117–123.
[20] A. Abraham, Intelligent systems: architectures and perspectives, recent advances in intelligent paradigms and applications, in: A. Abraham, L. Jain, J. Kacprzyk (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag, Germany, 2002, pp. 1–35 (Chapter 1).
[21] A. Abraham, M.R. Khan, Neuro-fuzzy paradigms for intelligent energy management, innovations in intelligent systems: design, management and applications, in: A. Abraham, L. Jain, B. Jan van der Zwaag (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag, Germany, 2003, pp. 285–314 (Chapter 12).
[22] D. Nauck, R. Kruse, NEFCLASS: a neuro-fuzzy approach for the classification of data, in: K. George et al. (Eds.), Proceedings of ACM Symposium on Applied Computing, ACM Press, Nashville, 1995, pp. 461–465.
[23] D. Nauck, R. Kruse, A neuro-fuzzy method to learn fuzzy classification rules from data, Fuzzy Sets and Systems 89 (1997) 277–288.
[24] D. Nauck, R. Kruse, Neuro-fuzzy systems for function approximation, Fuzzy Sets and Systems 101 (1999) 261–271.
[25] R.R. Yager, D.P. Filev, Adaptive defuzzicication for fuzzy system modeling, in: Proceedings of the Workshop of the North American Fuzzy Information Processing Society, 1992, pp. 135–142.
[26] Juan R. Castro, O. Castillo, P. Melin, Building fuzzy inference systems with the interval type-2 fuzzy logic toolbox, in: Proceedings of IFSA 2007, vol. 41, Part I, 2007, pp. 53–62.
[27] P. Werbos, Beyond regression: new tools for prediction and analysis in the behavioral sciences, Ph.D. Thesis, Harvard University, 1974.
[28] K.S. Narendra, K. Parthsarathy, Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks 1 (1) (1990) 4–27.
[29] M.T. Hagan, H.B. Demuth, M.H. Beale, Neural Network Design, PWS Publishing, Boston, MA, 1996.
[30] Roberto Sepúlveda, Oscar Castillo, Patricia Melin, Antonio Rodríguez Díaz, Oscar Montiel, Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic, Information Sciences 177 (10) (2007) 2023–2048.
[31] R. Sepulveda, O. Castillo, P. Melin, O. Montiel, An efficient computational method to implement type-2 fuzzy logic in control applications, in: Patricia Melin et al. (Eds.), Analysis and Design of Intelligent Systems using Soft Computing Techniques, Number 1 in Studies in Fuzziness and Soft Computing, first ed., vol. 5, Springer-Verlag, Germany, 2007, pp. 45–52.
[32] Ming-Ying Hsiao, Tzuu-Hseng S. Li, J.-Z. Lee, C.-H. Chao, S.-H. Tsai, Design of interval type-2 fuzzy sliding-mode controller, Information Sciences 178 (6) (2008) 1696–1716.
[33] Dongrui Wu, Jerry M. Mendel, Uncertainty measures for interval type-2 fuzzy sets, Information Sciences 77 (23) (2007) 5378–5393.
[34] Jerry M. Mendel, Hongwei Wu, New results about the centroid of an interval type-2 fuzzy set, including the centroid of a fuzzy granule, Information Sciences 177 (2) (2007) 360–377.
[35] Jerry M. Mendel, Advances in type-2 fuzzy sets and systems, Information Sciences 177 (1) (2007) 84–110.
[36] Patricia Melin, Oscar Castillo, An intelligent hybrid approach for industrial quality control combining neural networks, fuzzy logic and fractal theory, Information Sciences 177 (7) (2007) 1543–1557.
[37] Nilesh N. Karnik, Jerry M. Mendel, Applications of type-2 fuzzy logic systems to forecasting of time-series, Information Sciences 120 (1–4) (1999) 89–111.
[38] W. Pedrycz, J. Szymanski, S. Frimpong, A generalized granular multiplexer, in: International Conference on Applied Simulation and Modelling ASM 2000, July 24–26, Banff, 2000.
[39] W. Pedrycz, M.H. Smith, Information granulation: percepts and their stability, in: Proceedings of the IEEE Conference on Fuzzy Systems, San Antonio, 7–10 May, 2000, pp. 83–88.