Engineering Applications of Artificial Intelligence 71 (2018) 259–274

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A Beta basis function Interval Type-2 Fuzzy Neural Network for time series applications

N. Baklouti [a,*], A. Abraham [b], A.M. Alimi [a]

[a] REGIM-Lab, REsearch Groups in Intelligent Machines, University of Sfax, ENIS, BP 1173, Sfax, 3038, Tunisia
[b] Machine Intelligence Research Labs (MIR Labs), P.O. Box 2259, Auburn, Washington 98071-2259, USA

ARTICLE INFO

Keywords:
Beta function
Neural fuzzy network
Type-2 fuzzy logic
Mackey glass time series
ECG electrodiagram time series

ABSTRACT

The huge complexity and uncertainty in real life requires the use of advanced automatic learning methods to find out better approximators and suitable relationship in real data behavior. Neuro fuzzy systems have been proved to be excellent universal approximators. In this paper we propose a new based function Interval Type-2 Fuzzy Neural Network denoted ''Beta basis function Interval Type-2 Fuzzy Neural Network'', the BIT2FNN. The main idea is to involve type-2 beta fuzzy sets in the design process of fuzzy networks. The proposed architecture is based on beta type-2 fuzzy sets in the antecedent part, while the consequent part achieves the TSK (Takagi–Sugeno–Kang) fuzzy output strategy. Thanks to the beta function flexibility, the network achieve a good performance and shows a good resistance to noisy data. First order derivatives of type-1 and type-2 Beta functions were developed for the first time for designing fuzzy logic systems based on given input–output pairs. The backpropagation algorithm was used for the learning process of antecedent fuzzy beta parameters and the consequent part. The performance of the proposed model of Beta fuzzy logic system is evaluated with mainly two problems of time series applications : the Mackey Glass Chaotic Time-Series prediction problem with different setting of parameters and levels of noise and the ECG heart-rate Time Series monitoring problem.

1. Introduction

Fuzzy neural networks have been widely used in intelligent methodologies to settle serious data science problems, since they provide better learning capabilities. For instance, they have been fruitful applied in solving non linear and complex systems. Fuzzy systems (FSs) have been demonstrated to have good approximation capabilities (Wang, 1992), which have been used widely for approximating non linear functions and behaviors and forecasting many activities. Regarding learning type-1 fuzzy systems applied to regression, non-linear identification and time series problems, several works have been considered. In Fletcher and Reeves (1964), The neural network weights adaptation is proposed using an adaptive learning rate and momentum variable. An adaptive neural fuzzy inference system have been proposed first by Jang in 1998 (Jang, 1993) denoted and known by ANFIS. This Neural Fuzzy System (NFS) is based on an adaptive neural network structure using either the backpropagation algorithm or hybrid learning algorithm and a TSK model fuzzy logic system. In 1997, Mendel elaborated a type-1 FLS for the first time applied to the Mackey glass time series prediction problem (Mendel and Mouzouris, 1997). The TSK fuzzy structure was considered in several studies. In Jang (1993), Jang has defined an adaptive neural fuzzy inference system applied to the chaotic time series problem identification. For the learning process the gradient descent based approach was utilized. In Juang and Lin (1998), a SONFIN self constructing type-1 TSK FNN having an online learning ability was suggested using a competitive learning method. In Wu and Er (2000), a hierarchical on-line fuzzy neural network based radial basis function for TSK systems was proposed. Then through many theoretical studies and several successful applications, type-2 fuzzy neural systems have proved their effectiveness regarding type-1 FNNs. In 1999 (Karnik and Mendel, 1999), T2 concepts were added and applied for the Mackey glass time series prediction problem. In Liang and Mendel (2000), Liang and Mendel proposed a simplification of the generalized type-2 fuzzy set theory to introduce by this the concept of interval fuzzy sets. Since, much intention has been dutiful to learning type-2 FLS to have better approximation of non linear aptitudes (Das et al., 2015; Tung et al., 2013; Lee et al., 2003; Wang et al., 2004; Juang and Tsao, 2008). Jung in Juang and Tsao (2008), proposed an online structure of a TSK type-2 fuzzy neural network for fuzzy parameter learning. In this study, the antecedent fuzzy parameters were tuned using gradient descent algorithm while consequent parameters were tuned using a kalman filter

* Corresponding author.
E-mail addresses: nesrine.baklouti@ieee.org (N. Baklouti), ajith.abraham@ieee.org (A. Abraham), adel.alimi@ieee.org (A.M. Alimi).

https://doi.org/10.1016/j.engappai.2018.03.006
Received 29 May 2016; Received in revised form 15 March 2017; Accepted 9 March 2018
Available online 23 March 2018

algorithm. While in Juang et al. (2010), an interval type-2 fuzzy neural network (IT2FNN) combined with support vector regression structure (IT2FNN-SVR) was proposed. In Mndez and de los Angeles Hernandez (2009), a hybrid learning methodology has been developed for interval FNN. The back-propagation algorithm and the recursive orthogonal least squares algorithm, were both applied for adjusting respectively antecedent and consequent membership functions fuzzy parameters. In MaNdez and De Los Angeles Hernández (2013), Mendez proposed a hybrid learning algorithm for settling parameters of a non-singleton interval A2-C1 model type-2 TSK FLS system. In Das et al. (2015) a sequential meta-cognitive architecture of a learning algorithm has been introduced using a model of a TSK type-2 NFS. Generally, the performance of a neural network relies essentially on two matters to know the network architecture and the transfer function used in layers. Regarding the network structure the most used and simplest in neural fuzzy architectures is the feedforward network with a simple or hybrid backpropagation weight adjusting algorithm. In learning type-2 fuzzy neural networks, the backpropagation algorithm with its variations is the most used weight adjusting method for tuning fuzzy neural networks parameters. This algorithm consists on reducing through iterations, error between actual (FS) output network and a desired output. The initial use of this algorithm for a fuzzy system was in 1992 (Wang and Mendel, 1992), since, this algorithm have been proved its efficiency in several fuzzy neural leaning process. The backpropagation algorithm was defined for TSK interval type-2 FLS first in Mendel (2001a). Derivatives and formulas that are needed to accomplish this algorithm were provided in Mendel (2004). Those mathematical relations have been given for only gaussian membership functions with uncertain standard deviations or uncertain mean as in Rubio-Solis and Panoutsos (2015). Learning type-2 FLS parameters have been proposed at first in Lee et al. (2003), Mendel (2004), Wang et al. (2004) and Uncu and Turksen (2007). In Castro et al. (2009), three (IT2FNN)s structures were defined using gradient descent backpropagation with and without adaptive learning procedure. While in Wang et al. (2004) an IT2FNN was proposed using backpropagation algorithm. In Gaxiola et al. (2014b) a backpropagation leaning algorithm is used in the FNN with type-1 and type-2 triangular fuzzy weights. In Gaxiola et al. (2015), authors proposed a neural network with a generalized type-2 fuzzy weights denoted (NNGT2FW) and presented a comparison to the neural network with interval type-2 fuzzy weights (NNIT2FW). Analysis were ensured using prediction of Mackey Glass time series. In the considered neural network back-propagation algorithm is applied and the adaptation of its weight is ensured using generalized type-2 fuzzy inference systems. Regarding the transfer function used in layers, for fuzzy neural networks we mean the shape of membership function used for defining a fuzzy system. As it was noticed in Wang et al. (2004), since the variation of initial values of membership functions may effect the performance of the training process, by consequent also the used kind of shapes of membership function may also alter the performance result of training. The most considered membership function in literature is the gaussian membership function (Gaxiola et al., 2014a). But, this does not exclude the existence of triangular and trapezoidal membership functions in some works (Ishibuchi et al., 1993, 1995). In the context of comparing membership functions (MFs), in Olivas et al. (2014) authors presented a comparative study on the impact of the utilization of triangular and gaussian MFs in an IT2 fuzzy system for adjusting the Particle Swarm Optimization algorithm parameters. Considering type-2 fuzzy systems (T2FSs), the major cause of migration from type-1 (T1) to type-2 (T2) systems is particularly in the reduction of error and uncertainty that could be provided by T2FSs. But, the contradiction that arises here is : on the one hand, type-2 fuzzy systems which thanks to their type-2 membership functions that afforded the opportunity to handle uncertainties, but, on the other hand, almost all studies about, have neglected the possible impact of the chosen shape of the type-2 membership function on the further reduce of that uncertainty. Nevertheless it was mentioned in an earliest work (Wang, 1992) the possible impact of the shape of chosen membership function of a fuzzy system on the smoothness of the input–output surface. In this context, we demonstrated in this paper the great effect that can yield a rich membership function, such the beta function. This function was firstly proposed by Alimi in 1997 (Alimi, 1997a), as a transfer function in an artificial feedforward three layers neural network denoted the Beta Basis Function Neural Network (BBFNN). The beta function has several benefits comparing to the gaussian one, for instance it has the ability to provide more rich shapes specifically in points of view linearity, asymmetry and flexibility (Alimi et al., 2000; Alimi, 2000; Alimi et al., 2003). Successful applications have been utilized this function including classification and pattern recognition (Alimi, 1997b; Ltaief et al., 2012; Bezine et al., 2007, 2003), modeling of neural networks (Bouaziz et al., 2013, 2014), and time series forecasting (Baklouti et al., 2015). In this paper, a Beta basis function Interval Type-2 Fuzzy Neural Network architecture BIT2FNN was introduced. Throughout this architecture, an interval type-2 beta fuzzy set was defined and first order derivatives of both type-1 and type-2 beta sets were calculated. Based on a given data pairs and upon the backpropagation learning algorithm, weight adjusting and fuzzy parameters update of the antecedent and consequent part were performed. Comparison results were carried out using both beta and gaussian membership functions. The BIT2FNN model was tested using essentially two examples of time series applications: the Mackey Glass Chaotic Time-Series prediction application with different setting of parameters and levels of noise and the ECG heart-rate Time Series monitoring application. The obtained results with beta functions presented good performances. The paper is organized by following; Section 2 presented a description of the interval type-2 beta fuzzy set and properties including the beta primary MF with uncertain center. Next section, considered calculation details of first order derivatives of type-1 and type-2 beta functions. In Section 4 the structure of the BIT2FNN was detailed. Then it follows a description of the used backpropagation learning methodology. In Section 6, simulation studies are depicted. Finally the paper is concluded.

2. Interval type-2 beta fuzzy set

2.1. Type-1 beta fuzzy set

A type-1 beta fuzzy set in one dimensional case is the earlier defined beta membership function in Alimi (1997a) and expressed as follows,

$$\beta(x;p,q,x_0,x_1)=\begin{cases}\left(\dfrac{x-x_0}{c-x_0}\right)^p\left(\dfrac{x_1-x}{x_1-c}\right)^q & \text{if } x\in]x_0,x_1[\\ 0 & \text{elsewhere}\end{cases}\tag{1}$$

p,q,x_0 and x_1 are real values with $p,q>0$ and $x_1>x_0$. The beta function center is defined by $c=(px_1+qx_0)/p+q$ and its width by $\sigma=x_1-x_0$. The great significance and worth of the beta function relies mainly on its ability to approximate several functions such as triangular and gaussian functions (Alimi, 2003). Fig. 1 illustrates various beta functions with different parameters setting. In Alimi (2003), Alimi has demonstrated the ability of the beta function to approximate the gaussian function, in which he proves that for any given precision ϵ, and for any given gaussian function $Gauss(x;c,\sigma)$ it exists a beta function $\beta(x;p,q,x_0,x_1)$ that can approximates the gaussian one by an error of less than ϵ, and it is worth noting that the reverse is not true. $\beta(x;p,q,x_0,x_1)-Gauss(x;c,\sigma)<\epsilon$ for any $x\in\Re$. Eq. (1) leads to:

$$\beta(x;c,\sigma,p,q)=\begin{cases}\left(1+\dfrac{(p+q)(x-c)}{\sigma p}\right)^p\times\left(1-\dfrac{(p+q)(c-x)}{\sigma q}\right)^q & \text{if } x\in]c-\dfrac{\sigma p}{p+q},c+\dfrac{\sigma q}{p+q}[\\ 0 & \text{else.}\end{cases}\tag{2}$$

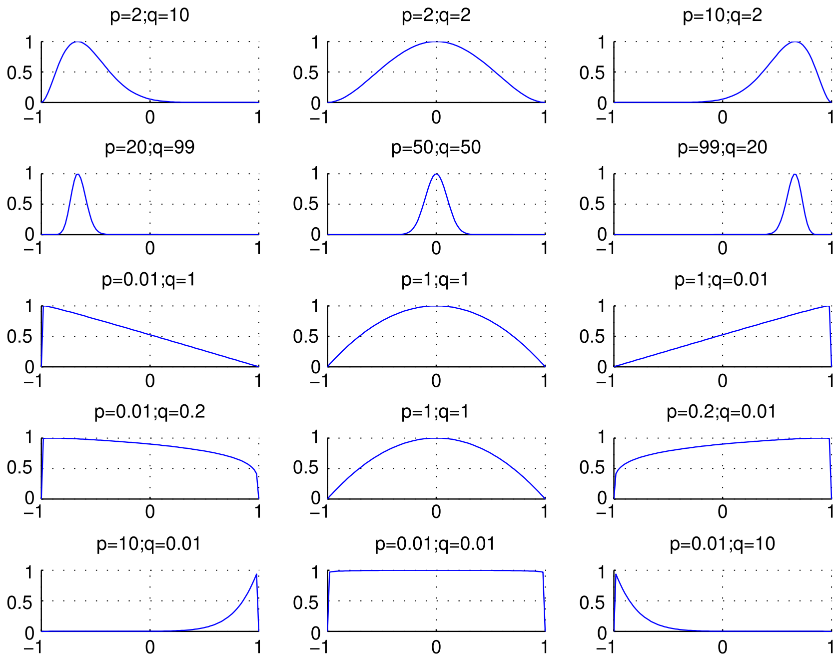


Fig. 1. Examples of type-1 beta membership functions.

2.2. *Interval type-2 beta fuzzy set: case of beta primary MF with uncertain center c*

An Interval beta type-2 (IT2) fuzzy set $\widetilde{B}$, such defined an interval type-2 gaussian set, is expressed by following equations (Mendel and John, 2002):

$$\widetilde{B} = \int_{x\in X}\int_{v\in J_X} 1/(x,v) \quad J_x \subseteq [0,1], \text{where every } \mu_{\widetilde{B}}(x,v)=1.$$
$$FOU(\widetilde{B}) = \bigcup_{x\in X} v \in J_x; \tag{3}$$
$$J_x = [\bar{\mu}_{\widetilde{B}}(x), \underline{\mu}_{\widetilde{B}}(x)], \forall x \in X.$$

The delimited region which is called the footprint of uncertainty (FOU), defines the uncertainty of the primary membership function (MF); where to each primary membership function exists a secondary 1-unit membership function. This bounded region is delimited by an (UMF) upper MF and a (LMF) lower MF called by $\bar{\mu}_{\widetilde{B}}(x)$ and $\underline{\mu}_{\widetilde{B}}(x)$, respectively. Fig. 2 illustrated different forms of type-2 beta functions with different parameter settings.

As the beta function has four parameters, then we can formulate the Interval type-2 Beta fuzzy set variously according to the parameter which is uncertain. In the whole paper we are using Beta primary MF with uncertain center c case. Other case studies are defined in the annex part.

Definition: beta primary MF with uncertain center *c:*. A beta primary membership function with uncertain center c, $c \in [c_1, c_2]$ is expressed as follows, whereabouts a different membership function matches to every c value.

$$\beta(x) = \left(1+\frac{(p+q)(x-c)}{\sigma p}\right)^p \left(1-\frac{(p+q)(c-x)}{\sigma q}\right)^q \quad \text{with } c \in [c_1,c_2]. \tag{4}$$

Fig. 3 shows an instance of this kind of type-2 fuzzy set.

Upper and lower MFs will then be calculated by the following Eqs. (5) and (6), respectively:

$$\bar{\mu}_{\widetilde{A}}(x) = \begin{cases} \beta(x;c_1,\sigma,p,q) & x < c_1 \\ 1 & c_1 < x < c_2 \\ \beta(x;c_2,\sigma,p,q) & x > c_2 \end{cases} \tag{5}$$

$$\underline{\mu}_{\widetilde{A}}(x) = \begin{cases} \beta(x;c_2,\sigma,p,q) & x \le (c_1+c_2)/2 \\ \beta(x;c_1,\sigma,p,q) & x > (c_1+c_2)/2. \end{cases} \tag{6}$$

3. First order derivatives of type-2 beta functions

Since our research study in this paper is the first which is defining type-2 fuzzy systems with beta membership functions, and for ensuring the training of type-1 and type-2 beta neural fuzzy systems using the back-propagation learning method, first order derivatives of type-1 and type-2 beta functions need to be elaborated.

3.1. *First order derivatives of type-1 beta functions*

For tuning the membership functions parameters of the antecedent and consequent of a type-1 beta FLS, we compute the following derivatives $\frac{\partial\mu_\beta(x)}{\partial c}$, $\frac{\partial\mu_\beta(x)}{\partial \sigma}$, $\frac{\partial\mu_\beta(x)}{\partial p}$ and $\frac{\partial\mu_\beta(x)}{\partial q}$ by the subsequent equations:

$$\frac{\partial\mu_\beta(x)}{\partial c} = \frac{\partial\beta(x)}{\partial c} = \frac{-(p+q)}{\sigma}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c-x)}{\sigma q}}\right) \tag{7}$$

$$\frac{\partial\mu_\beta(x)}{\partial \sigma} = \frac{\partial\beta(x)}{\partial \sigma} = \frac{(p+q)(c-x)}{\sigma^2}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c-x)}{\sigma q}}\right) \tag{8}$$

$$\frac{\partial\mu_\beta(x)}{\partial p} = \frac{\partial\beta(x)}{\partial p} = \beta(x)\times\left(\ln(1+\frac{(p+q)(x-c)}{\sigma p})\right) - \beta(x) \times\left(\frac{q(x-c)}{\sigma p+(p+q)(x-c)} - \frac{q(c-x)}{\sigma q-(p+q)(c-x)}\right) \tag{9}$$

$$\frac{\partial\mu_\beta(x)}{\partial q} = \frac{\partial\beta(x)}{\partial q} = \beta(x)\times\left(\ln(1-\frac{(p+q)(c-x)}{\sigma q})\right) - \beta(x) \times\left(\frac{p(c-x)}{\sigma p+(p+q)(x-c)} + \frac{p(c-x)}{\sigma q-(p+q)(c-x)}\right). \tag{10}$$

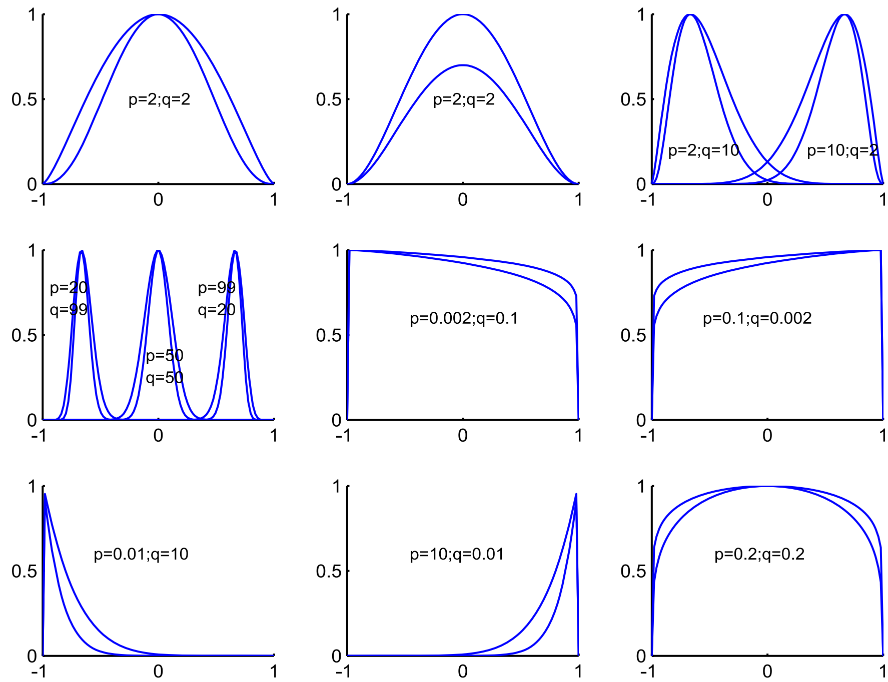


Fig. 2. Examples of type-2 beta membership functions.

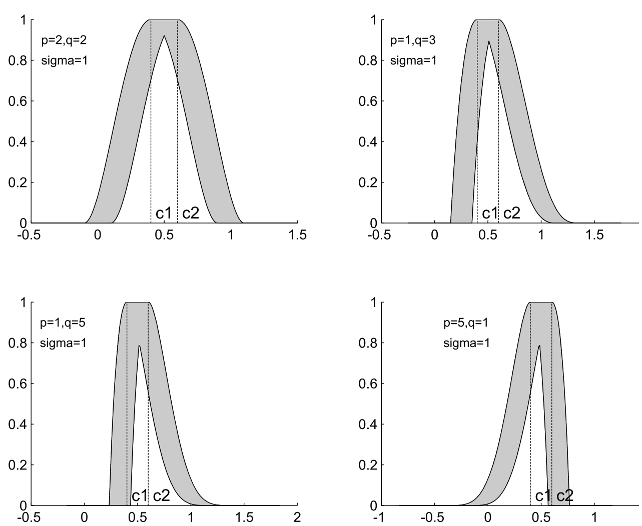


Fig. 3. Beta primary membership function with uncertain center.

3.2. Derivatives of beta primary MF with uncertain c

The antecedent and consequent MFs parameters' of the BIT2FNN fuzzy neural network are tuned by calculating derivatives of the primary beta membership functions. In the following, starting from Eqs. (5)–(6), and (7)–(10), the subsequent upper and lower derivatives, $\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial c_1}$, $\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial c_2}$, $\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial \sigma}$, $\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial p}$, $\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial q}$, $\frac{\partial\underline{\mu}_{\tilde{\beta}}(x)}{\partial c_1}$, $\frac{\partial\underline{\mu}_{\tilde{\beta}}(x)}{\partial c_2}$, $\frac{\partial\underline{\mu}_{\tilde{\beta}}(x)}{\partial \sigma}$, $\frac{\partial\underline{\mu}_{\tilde{\beta}}(x)}{\partial p}$ and $\frac{\partial\underline{\mu}_{\tilde{\beta}}(x)}{\partial q}$ are defined, where $x_I = (c_1 + c_2)/2$:

$$\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial c_1} = \begin{cases} \frac{-(p+q)}{\sigma}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c_1)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c_1-x)}{\sigma q}}\right) & x < c_1 \\ 0 & x \in [c_1, c_2] \\ 0 & c > c_2 \end{cases} \tag{11}$$

$$\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial c_2} = \begin{cases} 0 & x < c_1 \\ 0 & x \in [c_1, c_2] \\ \frac{-(p+q)}{\sigma}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c_2)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c_2-x)}{\sigma q}}\right) & x > c_2 \end{cases} \tag{12}$$

$$\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial \sigma} = \begin{cases} \frac{(p+q)(c_1-x)}{\sigma^2}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c_1)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c_1-x)}{\sigma q}}\right) & x < c_1 \\ 0 & c_1 < x < c_2 \\ \frac{(p+q)(c_2-x)}{\sigma^2}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c_2)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c_2-x)}{\sigma q}}\right) & x > c_2 \end{cases} \tag{13}$$

$$\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial p} = \begin{cases} \beta(x) \times \left[\left(\ln(1+\frac{(p+q)(x-c_1)}{\sigma p})\right) - \left(\frac{q(x-c_1)}{\sigma p+(p+q)(x-c_1)} - \frac{q(c_1-x)}{\sigma q-(p+q)(c_1-x)}\right)\right] & x < c_1 \\ 0 & c_1 < x < c_2 \\ \beta(x) \times \left[\left(\ln(1+\frac{(p+q)(x-c_2)}{\sigma p})\right) - \left(\frac{q(x-c_2)}{\sigma p+(p+q)(x-c_2)} - \frac{q(c_2-x)}{\sigma q-(p+q)(c_2-x)}\right)\right] & x > c_2 \end{cases} \tag{14}$$

$$\frac{\partial\bar{\mu}_{\tilde{\beta}}(x)}{\partial q} = \begin{cases} \beta(x) \times \left[\left(\ln(1-\frac{(p+q)(c_1-x)}{\sigma q})\right) - \left(\frac{p(c_1-x)}{\sigma p+(p+q)(x-c_1)} + \frac{p(c_1-x)}{\sigma q-(p+q)(c_1-x)}\right)\right] & x < c_1 \\ 0 & c_1 < x < c_2 \\ \beta(x) \times \left[\left(\ln(1-\frac{(p+q)(c_2-x)}{\sigma q})\right) - \left(\frac{p(c_2-x)}{\sigma p+(p+q)(x-c_2)} + \frac{p(c_2-x)}{\sigma q-(p+q)(c_2-x)}\right)\right] & x > c_2 \end{cases} \tag{15}$$

$$\frac{\partial\underline{\mu}_{\tilde{\beta}}(x)}{\partial c_1} = \begin{cases} 0 & x < x_I \\ \frac{-(p+q)}{\sigma}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c_1)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c_1-x)}{\sigma q}}\right) & x > x_I \end{cases} \tag{16}$$

$$\frac{\partial \mu_{\underline{\tilde{\beta}}}(x)}{\partial c_2} = \begin{cases} \frac{-(p+q)}{\sigma}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c_2)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c_2-x)}{\sigma q}}\right) & x > x_I \\ 0 & x < x_I \end{cases} \quad (17)$$

$$\frac{\partial \mu_{\underline{\tilde{\beta}}}(x)}{\partial \sigma} = \begin{cases} \frac{(p+q)(c_2-x)}{\sigma^2}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c_2)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c_2-x)}{\sigma q}}\right) \\ \frac{(p+q)(c_1-x)}{\sigma^2}\beta(x)\left(\frac{1}{1+\frac{(p+q)(x-c_1)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c_1-x)}{\sigma q}}\right) \end{cases} \quad (18)$$

$$\frac{\partial \mu_{\underline{\tilde{\beta}}}(x)}{\partial p} = \begin{cases} \beta(x)\times\left[\left(\ln(1+\frac{(p+q)(x-c_2)}{\sigma p})\right) - \left(\frac{q(x-c_2)}{\sigma p+(p+q)(x-c_2)} - \frac{q(c_2-x)}{\sigma q-(p+q)(c_2-x)}\right)\right] & x < x_I \\ \beta(x)\times\left[\left(\ln(1+\frac{(p+q)(x-c_1)}{\sigma p})\right) - \left(\frac{q(x-c_1)}{\sigma p+(p+q)(x-c_1)} - \frac{q(c_1-x)}{\sigma q-(p+q)(c_1-x)}\right)\right] & x > x_I \end{cases} \quad (19)$$

$$\frac{\partial \mu_{\underline{\tilde{\beta}}}(x)}{\partial q} = \begin{cases} \beta(x)\times\left[\left(\ln(1-\frac{(p+q)(c_2-x)}{\sigma q})\right) - \left(\frac{p(c_2-x)}{\sigma p+(p+q)(x-c_2)} + \frac{p(c_2-x)}{\sigma q-(p+q)(c_2-x)}\right)\right] & x < x_I \\ \beta(x)\times\left[\left(\ln(1-\frac{(p+q)(c_1-x)}{\sigma q})\right) - \left(\frac{p(c_1-x)}{\sigma p+(p+q)(x-c_1)} + \frac{p(c_1-x)}{\sigma q-(p+q)(c_1-x)}\right)\right] & x > x_I. \end{cases} \quad (20)$$

4. Beta basis function Interval Type-2 Fuzzy Neural Network (BIT2FNN)

The proposed architecture in our paper consists on the design of a MISO (multi-inputs/single-output) TSK (Takagi–Sugeno–Kang) (Takagi and Sugeno, 1985) Beta Basis Function Interval Type-2 Fuzzy Neural Network, which we note BIT2FNN. This model, point of view fuzzy system relies on an antecedent part based on beta type-2 membership functions. While consequent part is type-1 fuzzy sets based outputs.

A beta-TSK fuzzy basis system is defined by the up next relation (21):

$$f : I \subset \Re^n \to O \subset \Re^{n'}$$
$$y = f_j(x) = \frac{\sum_{j=1}^{M} f(x)\prod_{i=1}^{p}\mu^j_{\beta_i}(x_i)}{\sum_{k=1}^{M}\prod_{l=1}^{p}\mu^k_{\beta_l}(x_l)}. \quad (21)$$

In which: $\{I, O\}$ represents the $\{input, output\}$ space. $x = (x_1, x_2, \dots, x_p)^T \in I$ where $x_i \in [x_{i,min}, x_{i,max}]$. And $y \in O$. M represents the rule number. A rule is represented as follows: R_j:IF (x is β^j) Then ($y = f_j(x)$); $\beta^j = (\beta^j_1, \beta^j_2, \dots, \beta^j_p)^T$ specifies the linguistic fuzzy annotations assigned to beta membership functions $\mu^j_{\beta i}(x_i)$. f_j are the output polynomial functions with $I \subset \Re^n \to O \subset \Re$. f_j relies on input–output variables accordingly to the TSK model zero, first or second order. In this paper we are dealing with TSK type-2 fuzzy logic systems A2-C1 model. As reported in literature (Mendel, 2001b), this model is considering interval type-2 fuzzy sets and type-1 fuzzy sets in the antecedent and consequent parts, respectively. An A2-C1 TSK rule is represented by the following form:

$$\begin{aligned} &IF\ x_1 \text{ is } \tilde{F}^\ell_1\ and \dots and\ x_p\ is\ \tilde{F}^\ell_p\ \ THEN \\ &Y^\ell = C^\ell_0 + C^\ell_1 x_1 + \dots + C^\ell_p x_p \\ &C^\ell_j = [c^\ell_j - s^\ell_j, c^\ell_j - s^\ell_j] \end{aligned} \quad (22)$$

where c^ℓ_j and s^ℓ_j with $\ell = 1, \dots, M$ and $j = 0, \dots, p$, specified the center and spread of the interval type-1 consequent elements C^ℓ_j, respectively.

The proposed BIT2FNN is a six layered-type neural network. The whole architecture is depicted in Fig. 4. The process elaboration of those layers are given by the following.

4.0.1. Layer 1: input layer

That incorporates the vector input layer.

4.0.2. Layer 2: hidden beta basis function layer

This layer contains M nodes, each Mth one carries out the fuzzification process under the T2 beta MFs of the vector input data and yields to the upper and lower membership grades $\mu^{k\ell}_{\tilde{\beta}} = [\bar{\mu}^{k\ell}_{\tilde{\beta}}, \underline{\mu}^{k\ell}_{\tilde{\beta}}]$ of the kth feature of the ℓth rule, respectively. As described earlier in previous section, we used Beta primary MF with uncertain center:

$$\beta(x) = \left(1 + \frac{(p+q)(x-c)}{\sigma p}\right)^p \left(1 - \frac{(p+q)(c-x)}{\sigma q}\right)^q$$
$$with\ c \in [c_1, c_2]. \quad (23)$$

And the upper and the lower grades of the kth feature of the ℓth rule, are expressed by:

$$\bar{\mu}^{k\ell}_{\tilde{\beta}}(x) = \begin{cases} \beta^{k\ell}(x; c_1, \sigma, p, q) & x < c_1 \\ 1 & c_1 < x < c_2 \\ \beta^{k\ell}(x; c_2, \sigma, p, q) & x > c_2 \end{cases} \quad (24)$$

$$\underline{\mu}^{k\ell}_{\tilde{\beta}}(x) = \begin{cases} \beta^{k\ell}(x; c_2, \sigma, p, q) & x \le (c_1+c_2)/2 \\ \beta^{k\ell}(x; c_1, \sigma, p, q) & x > (c_1+c_2)/2. \end{cases} \quad (25)$$

4.0.3. Layer 3 (firing layer)

This layer is represented with M nodes. Each node defines the firing strength result of each of the M rules with meet operation under product t-norm, noted $[\bar{f}^\ell, \underline{f}^\ell]$. Then the output of a node is an interval type-1 set expressed as follows:

$$\begin{aligned} F^\ell(x') &= [\underline{f}^\ell(x'), \bar{f}^\ell(x')] \equiv [\underline{f}^\ell, \bar{f}^\ell] \\ &= [\underline{\mu}_{\tilde{F}^\ell_\ell}(x'_1) * \dots * \underline{\mu}_{\tilde{F}^\ell_p}(x'_p), \bar{\mu}_{\tilde{F}^\ell_\ell}(x'_1) * \dots * \bar{\mu}_{\tilde{F}^\ell_p}(x'_p)]. \end{aligned} \quad (26)$$

4.0.4. Layer 4

This layer is represented with M nodes which define the consequent nodes. Each node outcome from previous layer joined its associated consequent node in layer 4. This layer yields to the following interval type-1 fuzzy set:

$$\omega^\ell = [\omega^\ell_l, \omega^\ell_r] = [c^\ell_0 - s^\ell_0, c^\ell_0 + s^\ell_0] + \sum_{j=1}^{p}[c^\ell_j - s^\ell_j, c^\ell_j + s^\ell_j]x_j. \quad (27)$$

Then,

$$\omega^i_l = \sum_{j=0}^{p} c^\ell_j x_j - \sum_{j=0}^{p} x_j s^\ell_j\ where\ x_0 \triangleq 1. \quad (28)$$

And,

$$\omega^i_r = \sum_{j=0}^{p} c^\ell_j x_j + \sum_{j=0}^{p} x_j s^\ell_j\ where\ x_0 \triangleq 1. \quad (29)$$

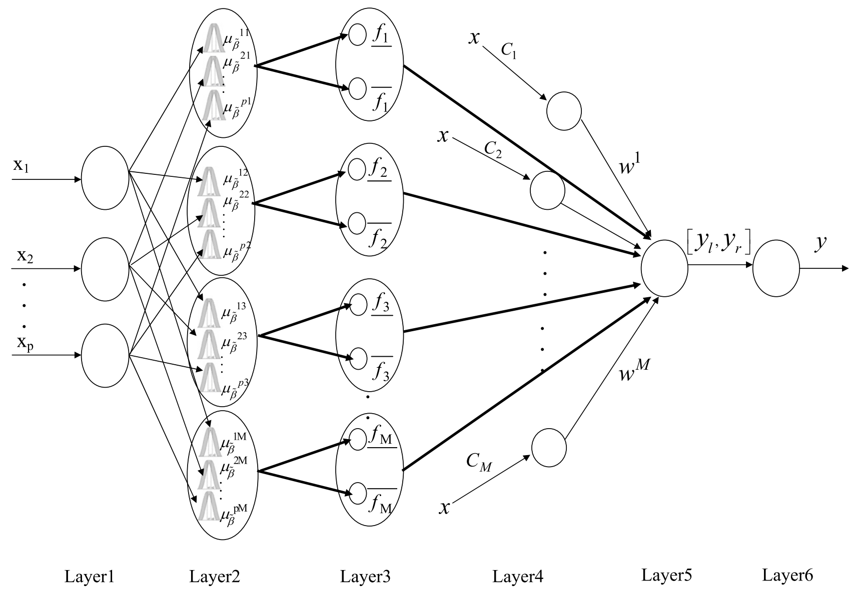


Fig. 4. BIT2FNN: A2-C1 TSK-type six layered neural network.

4.0.5. Layer 5

The output of this layer is an IT1 fuzzy set $[y_l, y_r]$ which indices j and r signify respectively the left and right limits. The calculation steps of this output can be based upon the Karnik and Mendel iterative procedure method (Mendel, 2001b). Explicitly in this technique, and based on [] the consequent values are in an ascending way reordered. Let we have the consequent value by the initial rule order ω_l and ω_r and the reordered sequence y_l and y_r expressed respectively by:

$\omega_l = (\omega_l^1, \dots, \omega_l^M)$ and $\omega_r = (\omega_r^1, \dots, \omega_r^M)$.

And
$y_l = (y_l^1, \dots, y_l^M)$ and $y_r = (y_r^1, \dots, y_r^M)$, in which $y_l^1 \le y_l^2 \le \cdots \le y_l^M$ and $y_r^1 \le y_r^2 \le \cdots \le y_r^M$.

According to Mendel the relation between the two vectors is presented by: $[y_l, y_r] = [Q_l\omega_l, Q_r\omega_r]$ where Q_l and Q_r are both elementary interchange matrices. Those matrices are used to reorder elements of the vectors ω_l and ω_r in an ascending order in respectively, the transformed vectors y_l and y_r. By reordering the $\bar{f}^{\ell}$ and $\underline{f}^{\ell}$ accordingly, we noted them $\bar{h}^{\ell}$ and $\underline{h}^{\ell}$, respectively. Then the outputs y_l and y_r can be calculated by the following equation:

$$y_l = \frac{\sum_{\ell=1}^{L} \bar{h}^{\ell} y_l^{\ell} + \sum_{\ell=L+1}^{M} \underline{h}^{\ell} y_l^{\ell}}{\sum_{\ell=1}^{L} \bar{h}^{\ell} + \sum_{\ell=L+1}^{M} \underline{h}^{\ell}}. \tag{30}$$

And

$$y_r = \frac{\sum_{\ell=1}^{L} \underline{h}^{\ell} y_r^{\ell} + \sum_{\ell=L+1}^{M} \bar{h}^{\ell} y_r^{\ell}}{\sum_{\ell=1}^{L} \underline{h}^{\ell} + \sum_{\ell=L+1}^{M} \bar{h}^{\ell}}. \tag{31}$$

4.0.6. Layer 6 (output layer)

Throughout this layer the output is computed by calculating the average of incoming data from the previous nodes. Then the final output y will be equal to:

$$y = (y_l + y_r)/2. \tag{32}$$

Then, according to the presented equations and anatomy, the final output is calculated depending on the weighting interval sets and the upper and lower antecedent membership functions. In next section, we depict the backpropagation learning algorithm applied throughout the BIT2FNN. The vital role of the algorithm is to reduce errors between the neural fuzzy system output and the desired output to find out the best relationship between data.

5. Backpropagation learning methodology

The beta fuzzy membership functions parameters are adjusted using backpropagation leaning algorithm. The error will be by then back propagated from the last layer until arriving to the first. And the antecedent and consequent parameters will be altered by the gradient descent method in accordance with the error between actual system output and the desired output. For a given input–output pairs $(x^i : y_d^i)$, the main idea is to find a FLS such that the error function given in Eq. (33) is minimized:

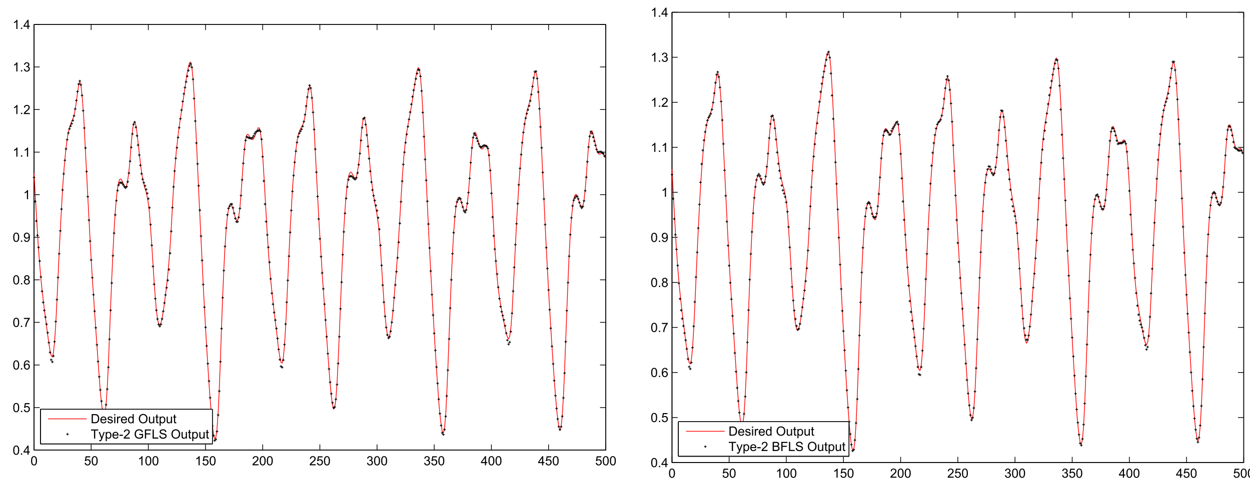
$$e^i = \frac{1}{2}[f_j(\underline{x})^i - y_d^i]^2 \quad i = 1, \dots, N. \tag{33}$$

Using (21) which depends on y^l and the beta parameters $p_{\beta_k^l}$, $q_{\beta_k^l}$, $x_{0\beta_k^l}$ and $x_{1\beta_k^l}$, where $l = 1, \dots, M$ represents the rule number and $k = 1, \dots, p$ the input index, we apply the steepest descent algorithm. This algorithm consists essentially in minimizing $J(\theta)$ in the following equation: $\theta(i+1) = \theta(i) - \alpha grad_{\theta}[J(\theta)]_i$ with α is the step size that belongs to $[0, 1]$. Based on this relation and since $J(\theta) = e^i$ in (33), we need to derive the parameters of both T1 and T2 beta fuzzy systems. For the update of type-1 beta parameters, the update equations of the antecedent and consequent parameters are calculated by next equations:

$$c_{\beta_k^l}(i+1) = c_{\beta_k^l}(i) - \alpha(f_s(x^{(i)}) - y_d^{(i)})(\bar{y}^l(i) - f_s(x^{(i)})) \times \frac{-(p+q)}{\sigma}\left(\frac{1}{1+\frac{(p+q)(x-c_1)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c_1-x)}{\sigma q}}\right)\phi_l(x^{(i)}) \tag{34}$$

$$\sigma_{\beta_k^l}(i+1) = \sigma_{\beta_k^l}(i) - \alpha(f_s(x^{(i)}) - y_d^{(i)})(\bar{y}^l(i) - f_s(x^{(i)})) \times \frac{(p+q)(c-x)}{\sigma^2}\left(\frac{1}{1+\frac{(p+q)(x-c)}{\sigma p}} + \frac{1}{1-\frac{(p+q)(c-x)}{\sigma q}}\right)\phi_l(x^{(i)}) \tag{35}$$

$$p_{\beta_k^l}(i+1) = p_{\beta_k^l}(i) - \alpha(f_s(x^{(i)}) - y_d^{(i)})(\bar{y}^l(i) - f_s(x^{(i)})) \times \left[\left(\ln(1+\frac{(p+q)(x-c)}{\sigma p})\right) - \left(\frac{q(x-c)}{\sigma p+(p+q)(x-c)} - \frac{q(c-x)}{\sigma q-(p+q)(c-x)}\right)\right]\phi_l(x^{(i)}) \tag{36}$$



(a) (b)

Fig. 5. Output and desired output for: (a) GIT2FNN, (b) BIT2FNN.

$$q_{\beta_k^l}(i+1) = q_{1\beta_k^l}(i) - \alpha(f_s(x^{(i)}) - y_d^{(i)})(\bar{y}^l(i) - f_s(x^{(i)})) \times \left[\left(\ln(1 - \frac{(p+q)(c-x)}{\sigma q})\right) - \left(\frac{p(c-x)}{\sigma p + (p+q)(x-c)} + \frac{p(c-x)}{\sigma q - (p+q)(c-x)}\right)\right] \phi_l(x^{(i)}). \quad (37)$$

Regarding the type-2 beta neuro-fuzzy system, BIT2FNN, which is characterized by an uncertain center, the goal of the backpropagation algorithm is to estimate the antecedent and the consequent parameters giving the lowest error, to listing c_{k1}^ℓ, c_{k2}^ℓ, σ_k^ℓ, p_k^ℓ, q_k^ℓ and $[w_l^\ell, w_r^\ell]$. Thereby we derive the subsequent equations:

$$c_k^\ell(i+1) = c_k^\ell(i) - \alpha \frac{\partial(e^i)}{\partial(c_k^\ell)}|_i$$
$$c_k^\ell(i+1) = c_k^\ell(i) - \frac{1}{2}\alpha(f_j(\underline{x})^i - y_d^i) \times [\frac{w_{lr}^\ell - y_{lr}}{\prod_{k=1}^N \mu_{F_k^\ell}}] \frac{\partial \mu_\beta(x)}{\partial c_k^\ell(i)}$$
$$c_k^\ell(i+1) = c_k^\ell(i) - \frac{1}{2}\alpha(f_j(\underline{x})^i - y_d^i) \times [\frac{w_{lr}^\ell - y_{lr}}{\prod_{k=1}^N \mu_{F_k^\ell}}] \times \beta(x_k) \times \frac{-(p_k^\ell + q_k^\ell)}{\sigma_k^\ell}\left(\frac{1}{1 + \frac{(p_k^\ell+q_k^\ell)(x_k - c_k^\ell)}{\sigma_k^\ell p_k^\ell}} + \frac{1}{1 - \frac{(p_k^\ell+q_k^\ell)(c_k^\ell - x_k)}{\sigma_k^\ell q_k^\ell}}\right) \quad (38)$$

where $c_k^\ell \in [c_{k1}^\ell, c_{k2}^\ell]$.

Likewise, for tuning σ, p, q and the weighting parameters, we derive the following equations:

$$\sigma_k^\ell(i+1) = \sigma_k^\ell(i) - \alpha \frac{\partial(e^i)}{\partial(\sigma_k^\ell)}|_i$$
$$\sigma_k^\ell(i+1) = \sigma_k^\ell(i) - \frac{1}{2}\alpha(f_j(\underline{x})^i - y_d^i) \times [\frac{w_{lr}^\ell - y_{lr}}{\prod_{k=1}^N \mu_{F_k^\ell}}] \frac{\partial \mu_\beta(x)}{\partial \sigma_k^\ell(i)}$$
$$\sigma_k^\ell(i+1) = \sigma_k^\ell(i) - \frac{1}{2}\alpha(f_j(\underline{x})^i - y_d^i) \times [\frac{w_{lr}^\ell - y_{lr}}{\prod_{k=1}^N \mu_{F_k^\ell}}] \times \beta(x_k) \times \frac{(p_k^\ell + q_k^\ell)(c_k^\ell - x_k)}{\sigma_k^{\ell 2}}\left(\frac{1}{1 + \frac{(p_k^\ell+q_k^\ell)(x_k - c_k^\ell)}{\sigma_k^\ell p_k^\ell}} + \frac{1}{1 - \frac{(p_k^\ell+q_k^\ell)(c_k^\ell - x_k)}{\sigma_k^\ell q_k^\ell}}\right) \quad (39)$$

$$p_k^\ell(i+1) = p_k^\ell(i) - \alpha \frac{\partial(e^i)}{\partial(p_k^\ell)}|_i$$
$$p_k^\ell(i+1) = p_k^\ell(i) - \frac{1}{2}\alpha(f_j(\underline{x})^i - y_d^i) \times [\frac{w_{lr}^\ell - y_{lr}}{\prod_{k=1}^N \mu_{F_k^\ell}}] \frac{\partial \mu_\beta(x)}{\partial p_k^\ell(i)}$$
$$p_k^\ell(i+1) = p_k^\ell(i) - \frac{1}{2}\alpha(f_j(\underline{x})^i - y_d^i) \times [\frac{w_{lr}^\ell - y_{lr}}{\prod_{k=1}^N \mu_{F_k^\ell}}] \times \beta(x_k) \times \left[\left(\ln(1 + \frac{(p_k^\ell + q_k^\ell)(x_k - c_k^\ell)}{\sigma_k^\ell p_k^\ell})\right) - \left(\frac{q_k^\ell(x_k - c_k^\ell)}{\sigma_k^\ell p_k^\ell + (p_k^\ell + q_k^\ell)(x_k - c_k^\ell)} - \frac{q_k^\ell(c_k^\ell - x_k)}{\sigma_k^\ell q_k^\ell - (p_k^\ell + q_k^\ell)(c_k^\ell - x_k)}\right)\right] \quad (40)$$

$$q_k^\ell(i+1) = q_k^\ell(i) - \alpha \frac{\partial(e^i)}{\partial(q_k^\ell)}|_i$$
$$q_k^\ell(i+1) = q_k^\ell(i) - \frac{1}{2}\alpha(f_j(\underline{x})^i - y_d^i) \times [\frac{w_{lr}^\ell - y_{lr}}{\prod_{k=1}^N \mu_{F_k^\ell}}] \frac{\partial \mu_\beta(x)}{\partial q_k^\ell(i)}$$
$$q_k^\ell(i+1) = q_k^\ell(i) - \frac{1}{2}\alpha(f_j(\underline{x})^i - y_d^i) \times [\frac{w_{lr}^\ell - y_{lr}}{\prod_{k=1}^N \mu_{F_k^\ell}}] \times \beta(x_k) \times \left[\left(\ln(1 - \frac{(p_k^\ell + q_k^\ell)(c_k^\ell - x_k)}{\sigma_k^\ell q_k^\ell})\right) - \left(\frac{p_k^\ell(c_k^\ell - x_k)}{\sigma_k^\ell p_k^\ell + (p_k^\ell + q_k^\ell)(x_k - c_k^\ell)} + \frac{p_k^\ell(c_k^\ell - x_k)}{\sigma_k^\ell q_k^\ell - (p_k^\ell + q_k^\ell)(c_k^\ell - x_k)}\right)\right] \quad (41)$$

$$w_{lr}^\ell(i+1) = w_{lr}^\ell(i) - \frac{1}{2}\alpha(f_j(\underline{x})^i - y_d^i) \times \frac{\beta(x)}{\prod_{k=1}^N \mu_{F_k^\ell}}. \quad (42)$$

In next section, we carried out simulation analysis with mainly two examples of time series applications: the Mackey Glass Chaotic Time-Series prediction problem with different setting of parameters and levels of noise and the ECG heart-rate Time Series monitoring problem.

6. Simulation studies

6.1. Example 1 (a): free noise Mackey Glass Chaotic Time-Series prediction

To evaluate the performance of Beta fuzzy sets we test the well known benchmark of Forecasting of Time-Series. The considered problem is an important case study that appears in many analysis, where the main idea is better weather forecasts can, better the return on an

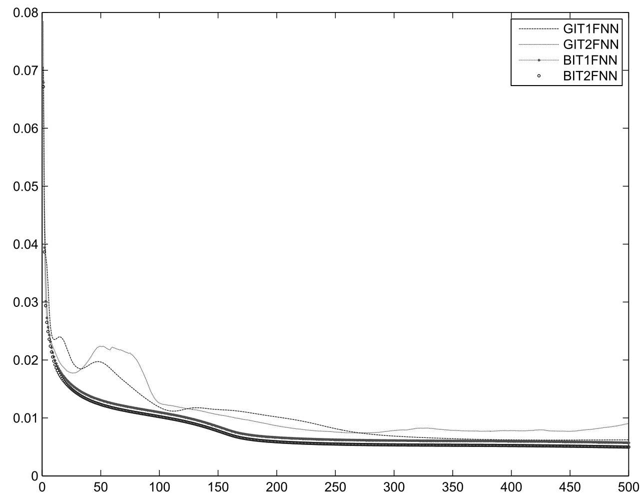


Fig. 6. RMSE of different FLS with beta or gaussian MF for 500 iterations.

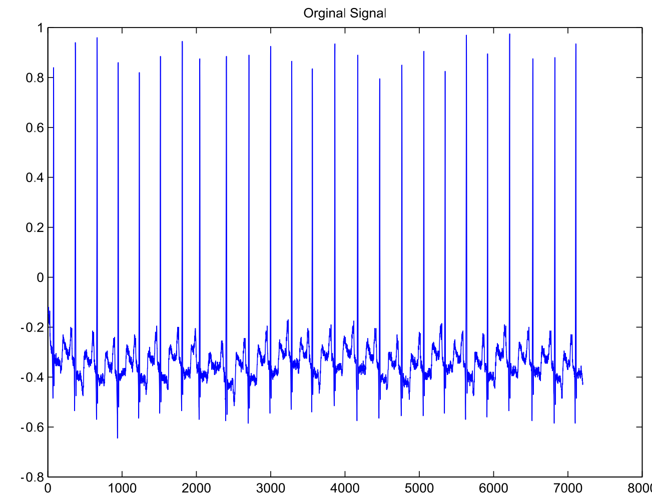
investment can occurs. In this part we are considering a free noise data. The Mackey Glass Chaotic Time-Series data can be generated using the delay differential equation:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t). \tag{43}$$

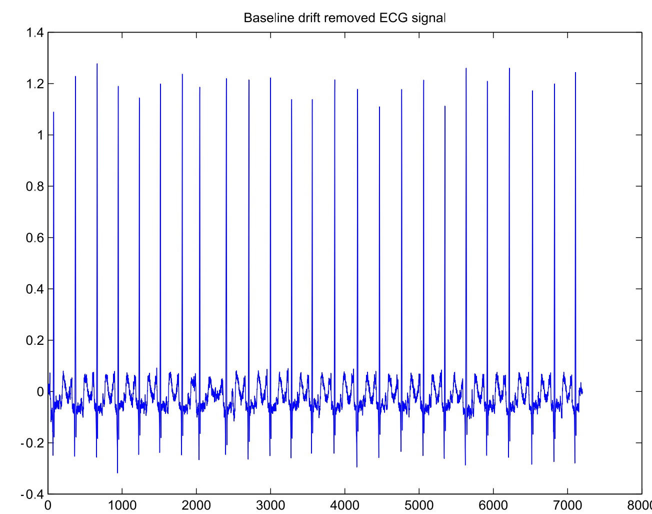
The setting of parameters of the data changes from one work to another. In this part, we adopted the most used settings parameters, as in Gaxiola et al. (2015), Gaxiola et al. (2014b) and Zarandi et al. (2013), we take $a = 0.2$, $b = 0.1$, $c = 10$ and $\tau = 17$. In this forecasting problem, four past values are used to predict a current value $x(t)$. The data pairs are defined by: $[inputs; outputs] = [x(t-24), x(t-18), x(t-12), x(t-6); x(t)]$.

We firstly design and fix the architecture of the FLS ahead the time, then we use the data training pairs for optimizing the membership function parameters. At first, we choose to make a comparative analysis using both, the gaussian and the beta, shapes of membership functions. We considered essentially four fuzzy neural networks using the same backpropagation learning algorithm with type-1 and type-2 fuzzy sets and using the both forms of membership functions. We will adopt abbreviated names to design the four systems which are depending on the nature of FLS itself and used MF within, i.e type-1 or type-2 and the used membership functions gaussian functions or beta functions. The name BIT2FNN designs our proposed system, Beta Basis Function Interval Type-2 Fuzzy Neural Network, while GIT2FNN designs a type-2 FNN using gaussian membership functions. And we define by BT1FNN (Beta Type-1 Fuzzy Neural Network) and GT1FNN (Gaussian Type-1 Fuzzy Neural Network) the same considered systems, respectively, except that they are using type-1 membership functions instead type-2.

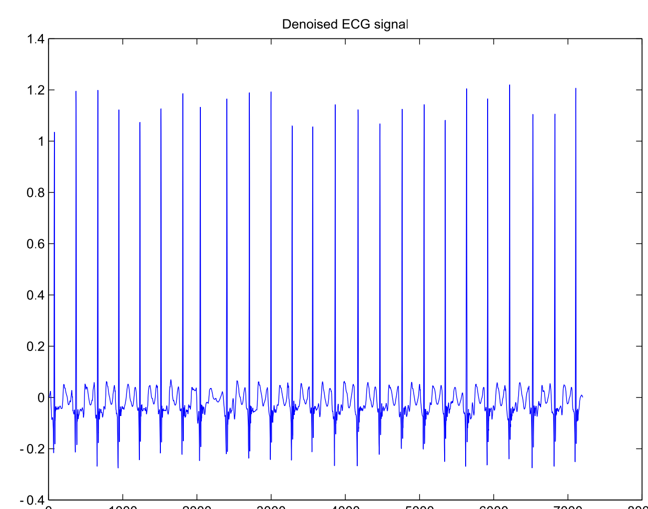
Regarding type-2 fuzzy systems, BIT2FNN and GIT2FNN are singleton TSK FLS zero order (A2-C1), where membership functions in the antecedent part are type-2 fuzzy sets and in the consequent part are interval type-1 fuzzy sets. All considered systems have four inputs, each one with two membership functions, and one output (the MISO case: Multi Inputs Single Output). Reduction of number of rules is not considered in this work. We envisage using the same maximum number of rules for the four considered models, which is equal to 16 rules to fairly compare the performances design under same conditions. In this section, we used data noise free, which are denoted $s(k)$. We designed the first 500 patterns $s(1001), \ldots s(1500)$ for the training phase and the remaining 500 patterns for testing the output design. Number of epoch for the convergence of the system was chosen equal to 1000. This value was essentially chosen a little big to guarantee the convergence of the FNN and to have a better comparison. Since the considered learning process is offline, optimization of number of rules or time convergence



(a) 100 ECG original signal.



(b) Baseline removed drift signal.



(c) The denoised signal.

Fig. 7. 100 ECG original, baseline removed drift and de-noised signals.

do not matter a lot regarding the error system output. Performance is evaluated using the root mean square error (RMSE) expressed as

Table 1
Comparison for 1000 iterations in Example 1.

Model	RMSE training	RMSE testing	Spread for T2 FLS
GT1FNN	0.0064	0.0062	–
GIT2FNN	0.0054	0.0055	Spread for antecedent = 0.1;
BT1FNN	0.0041	0.0042	–
BIT2FNN	0.0032	0.0034	Spread for consequent = 0.1

follows:

$$RMSE = \frac{1}{500}\sum_{k=1}^{500}[y(t+1) - y_d(t+1)]^2. \tag{44}$$

The α parameter was taken equal to 0.2. Considering the parameter initialization of the BT1FNN, initial values of standard deviation σ, p, q and center c were fixed to respectively 1.5086, 2, 2 and 0.4256. Regarding the BIT2FNN, where we are using beta primary membership functions with uncertain center, we initialized c_1 and c_2 values to respectively $c - 0.2$ and $c + 0.2$, in which c is randomly initialized.

In Fig. 6, the obtained root mean square error from the different systems are printed out. Those results are expressed in Table 1, where the RMSE values for each of the four systems are given. We can observe that fuzzy logic systems with either type-1 or type-2 beta membership functions gave better results with the lowest errors with regard to their gaussian membership functions systems. Final obtained forecasters for systems GIT2FNN and BIT2FNN are presented in Fig. 5(a) and (b), respectively, where both system outputs and the desired output are illustrated. Final values of the best obtained fuzzy neural network forecaster are presented in Tables 2–5.

In Table 6, we presented a comparison across our proposed beta fuzzy neural networks BIT2FNN and the BIT1FNN, and some other fuzzy neural networks known in literature. Those works are including the NN with generalized type-2 fuzzy weights (NNGT2FW) (Gaxiola et al., 2015), the NN with interval type-2 fuzzy weights (NNIT2FW) (Gaxiola et al., 2015) and the monolithic neural network (NN) (Zarandi et al., 2013). The associated simulation results were obtained from Gaxiola et al. (2014b).

As can be seen from the presented results, fuzzy neural networks using beta membership functions showed good performances. It can be concluded that beta type-2 neuro-fuzzy systems provide excellent forecasters and good relationship between input and output data for Mackey Glass Chaotic Time-Series problems.

6.2. Example 1 (b): noisy Chaotic Time-Series prediction

Since operating in a noise free environment does not represent the reality of things in real world, in this part we are considering a noisy environment to see how robust are the beta type-1 and type-2 forecasters. The noise ability of the proposed beta system is analyzed using different levels of noisy training and testing data. At first, We take original training data $x(t)$ while for testing data, we add a gaussian noise with a zero mean and standard deviations (SD) of 0.1 and 0.3, respectively. Results are summarized in Table 7. Then, we take training noisy data by adding to original data $x(t)$ a gaussian noise with a zero mean and standard deviations (SD) of 0.1, 0.2 and 0.3, respectively. Each of them was tested on three other datasets which are clean of noise, and with SD noise of 0.1 and 0.3, respectively. Results are showed in Table 8. The considered tables presented the training and

Table 2
Final values of p and q antecedent parameters for BIT2FNN in Example 1.

Rule N.	Final p values in antecedents				Final q values in antecedents			
1	2.0231	2.0049	1.9853	2.0143	1.9336	1.9813	1.9910	1.9617
2	1.9631	1.9828	1.9878	2.0154	2.1338	2.0590	2.0435	2.0890
3	1.9855	2.0177	2.0359	2.0184	2.0238	1.9850	1.8801	1.9420
4	2.0152	2.0107	1.9828	1.9838	1.9455	1.9120	1.8808	1.9224
5	1.9947	2.0125	1.9969	1.9958	2.0232	2.0067	2.0113	2.0120
6	2.0019	1.9981	2.0022	1.9917	1.9760	1.9523	1.9931	1.9986
7	1.9979	1.9964	1.9976	1.9863	2.0211	2.0313	2.0521	2.0433
8	2.0619	2.0368	1.9901	1.9412	1.9321	2.1824	1.9405	1.9616
9	1.9527	2.0417	2.0814	2.0396	1.8759	1.8739	1.8253	1.8778
10	1.9892	1.9949	1.9999	1.9998	2.0143	2.0165	2.0048	2.0046
11	1.9755	2.0053	2.0005	2.0114	2.0133	1.9857	1.9812	1.9552
12	2.0262	2.0050	2.0118	1.9793	1.9518	1.9722	1.9746	2.0100
13	2.0284	1.9896	1.9981	1.9926	1.9996	2.0355	2.0136	2.0204
14	2.0203	2.0039	1.9996	1.9840	1.9735	2.0037	2.0108	2.0332
15	2.0554	2.0780	1.9165	2.1136	1.8920	1.7919	2.0978	1.7966
16	1.9876	2.0866	2.0278	1.8693	2.2511	2.3811	2.2484	2.3053

Table 3
Final center c, antecedent parameters values, for BIT2FNN in Example 1.

Rule N.	Final c values in antecedents							
	c_1	c_2	c_1	c_2	c_1	c_2	c_1	c_2
1	0.3982	0.5210	0.4143	0.4852	0.4143	0.4852	0.4582	0.4428
2	0.2260	0.3451	0.3583	0.38971	0.4016	0.5412	1.0853	1.1574
3	0.3145	0.4523	0.4083	0.5649	1.1034	1.4423	0.4919	0.8119
4	0.4355	0.5840	0.5638	0.6791	0.9820	1.127	0.9524	1.1890
5	0.3814	0.4527	1.0833	1.2843	0.4087	0.6472	0.4164	0.8576
6	0.5260	0.8691	1.1666	0.2891	0.4624	0.9531	0.9940	1.1120
7	0.3972	0.8614	0.9861	1.2161	1.0263	1.2237	0.4045	0.6241
8	0.6521	0.8542	0.8389	0.9215	0.9087	1.1201	0.8084	1.1871
9	0.8724	0.9543	0.5449	0.6517	0.6916	0.7465	0.5827	0.6742
10	0.9794	1.2451	0.3796	0.5278	0.3968	0.5345	0.9960	1.1042
11	0.9337	1.2103	0.3874	0.4256	0.9917	1.1420	0.4688	0.5214
12	1.1835	1.3591	0.5022	0.7153	1.1036	1.2459	0.9488	1.3245
13	1.1018	1.3210	0.9076	1.1389	0.4030	0.6214	0.3953	0.5234
14	1.1034	1.2358	1.0679	1.2410	0.3958	0.5219	0.9724	1.2140
15	1.0344	1.8941	1.1637	1.9125	0.8167	0.9872	0.6375	0.8654
16	0.8357	0.9823	1.2657	1.5634	1.0512	1.6534	0.7907	0.9853

Table 4
Final sigma σ, antecedent parameters values, for BIT2FNN in Example 1.

Rule N.	Final sigma values in antecedents			
1	1.5554	1.4506	1.5622	1.4975
2	1.2754	1.4558	1.5223	0.8964
3	1.3494	1.3687	0.7921	1.5651
4	1.4593	1.7576	1.2739	0.9303
5	1.4417	0.8772	1.4889	1.5091
6	1.7060	0.7098	1.5703	0.9726
7	1.4604	1.0628	0.9113	1.5239
8	1.6136	0.9138	1.1396	1.3515
9	1.1656	1.5712	1.6131	1.6424
10	1.0142	1.4383	1.4518	0.9936
11	0.9898	1.4042	1.0087	1.5499
12	0.7632	1.6440	0.8822	1.1339
13	0.8611	1.0686	1.4768	1.4851
14	0.9301	0.8880	1.4496	0.9806
15	0.6340	1.0456	1.2718	1.4631
16	1.1314	0.7132	0.8688	1.4844

Table 5
Final consequent parameters for the BIT2FNN in Example 1.

Rule N.	Consequent final values	
1	0.4170	0.5096
2	0.5458	0.6672
3	0.0119	0.0145
4	0.9578	1.1706
5	0.6841	0.8361
6	0.0867	0.1059
7	0.4648	0.5680
8	1.3863	1.6943
9	1.2032	1.4706
10	0.8699	1.0633
11	0.2453	0.2999
12	0.7393	0.9035
13	0.5036	0.6156
14	0.4930	0.6026
15	0.2155	0.2635
16	0.7881	0.9633

Table 6
Comparison for 100 iterations in Example 1.

Model	RMSE testing
NN (Zarandi et al., 2013)	0.0530
NNGT2FW (Gaxiola et al., 2015)	0.0548
NNIT2FW (Gaxiola et al., 2015)	0.0355
BT1FNN	0.0110
BIT2FNN	0.0102

Table 7
Comparison with free Training noise (STD = 0) in Example 2.

FLS		GT1FNN	GIT2FNN	BT1FNN	BIT2FNN
Test RMSE	Free noise	0.0062	0.0055	0.0042	0.0034
	STD = 0.1	0.0092	0.0080	0.0053	0.0044
	STD = 0.3	0.0641	0.0533	0.0246	0.0126

testing RMSE for both type-2 gaussian and beta neural systems with the considered levels of noise. It can be seen from the established values that RMSE increases as the noise STD increases. But we can observe that the increase for $BIT2FNN$ is not significantly higher compared to GIT2FNN, showing in consequence the noise resistance ability of the proposed beta fuzzy network.

Simulation results showed that, compared with gaussian fuzzy systems, the Beta type-2 systems presented higher performance for both clean and noisy data. The major reason is that Beta function has the advantage to be flexible since it has the capacity to generate rich shapes (linearity, asymmetry, etc.) by this function add extra degree of freedom to the FOU zone.

Table 8
Prediction using noisy training and testing data in Example 2.

STD noise		G2FLS	B2FLS
Training	Testing	Test RMSE	Test RMSE
0.1	Free	0.0981	0.0439
	0.1	0.1583	0.0821
	0.3	0.2145	0.1657
0.2	Free	0.1325	0.056
	0.1	0.2025	0.087
	0.3	0.2984	0.180
0.3	Free	0.1836	0.092
	0.1	0.2871	0.127
	0.3	0.3411	0.182

In order to further study the importance of the beta function as a membership function in fuzzy systems, we test the ECG heart-rate Time Series monitoring problem.

6.3. Example 2: ECG heart-rate time series monitoring

In this part we consider the electrodiagram (ECG) time series application. ECG signals represent recordings of the electrical activity of the cardiac system. An ECG waveform signal consists essentially by the ventricular depolarization and the re-polarization of ventricle. In heart diseases, this signal is very important clinical information that must be controlled continuously. A normal ECG signal is always defined by regular patterns showing heart muscles' contractions and relaxations. In the case of abnormalities and heart diseases, waveform patterns will be altered from normal conditions to have irregular rhythms. This irregularity may be presented by too slow, too fast or altered waveforms. However, cardiac time series of normal and abnormal signals may be very similar. It is very important to automatically distinguish between them to have a better evaluation of abnormal situations, particularly those that may lead to the cardiac deaths. Hence, early detection of abnormal patterns in an ECG signal can alert about a heart disease and may even help to avoid a sudden cardiac death. People with heart problems must have a continuously monitoring of their heart's electrical activity. By consequent, the automation process of the monitoring task has become a strong necessity in last years. ECG signal processing materials deal with the use of many techniques as autocorrelation, wavelet transform parameter extraction, frequency analysis, baseline correction, de-noising (Mjahad et al., 2017; de Chazal et al., 2000; Seena and Yomas, 2014; Su and Zhao, 2005). In Xiong et al. (2016) for example, authors proposed a neural network for de-noising an ECG signal and removing any residual noise. The proposed method was performed on ECG recordings from both MIT-BIH Arrhythmia and Noise-Stress Test databases. Indeed, the ECG signal analysis have been discussed in several works providing a prediction of abnormalities and arrhythmias in the ECG signal. In Chua and Tan (2011) a non singleton genetic fuzzy logic system was proposed for the classification of arrhythmias in ECG signals from normal signals, signals with ventricular fibrillation or tachycardia, while in Mjahad et al. (2017) for the same problem classification authors used a time frequency approach. In Rai et al. (2013), authors combined multi-resolution wavelet transform and a neural network classifier for arrhythmia detection in an ECG signal. In this work 48 features have been extracted for each ECG signal. Those features have been defined based on Morphology features and discrete Wavelet Transform. Fuzzy neural networks have been used in some works for the ECG classification problem. In Gler and Beyli(2004) an adaptive neuro fuzzy system for screening ECG modifications for persons with partial epilepsy was explored. In Ranganathan et al. (2012) an estimation of heart-rate signals in different stress conditions was analyzed by using a neural fuzzy technique. Whereas only few works proposed type-2 fuzzy neural networks for resolving ECG arrhythmia detection and classification problems. The works in Zbay et al. (2011) and Ceylan

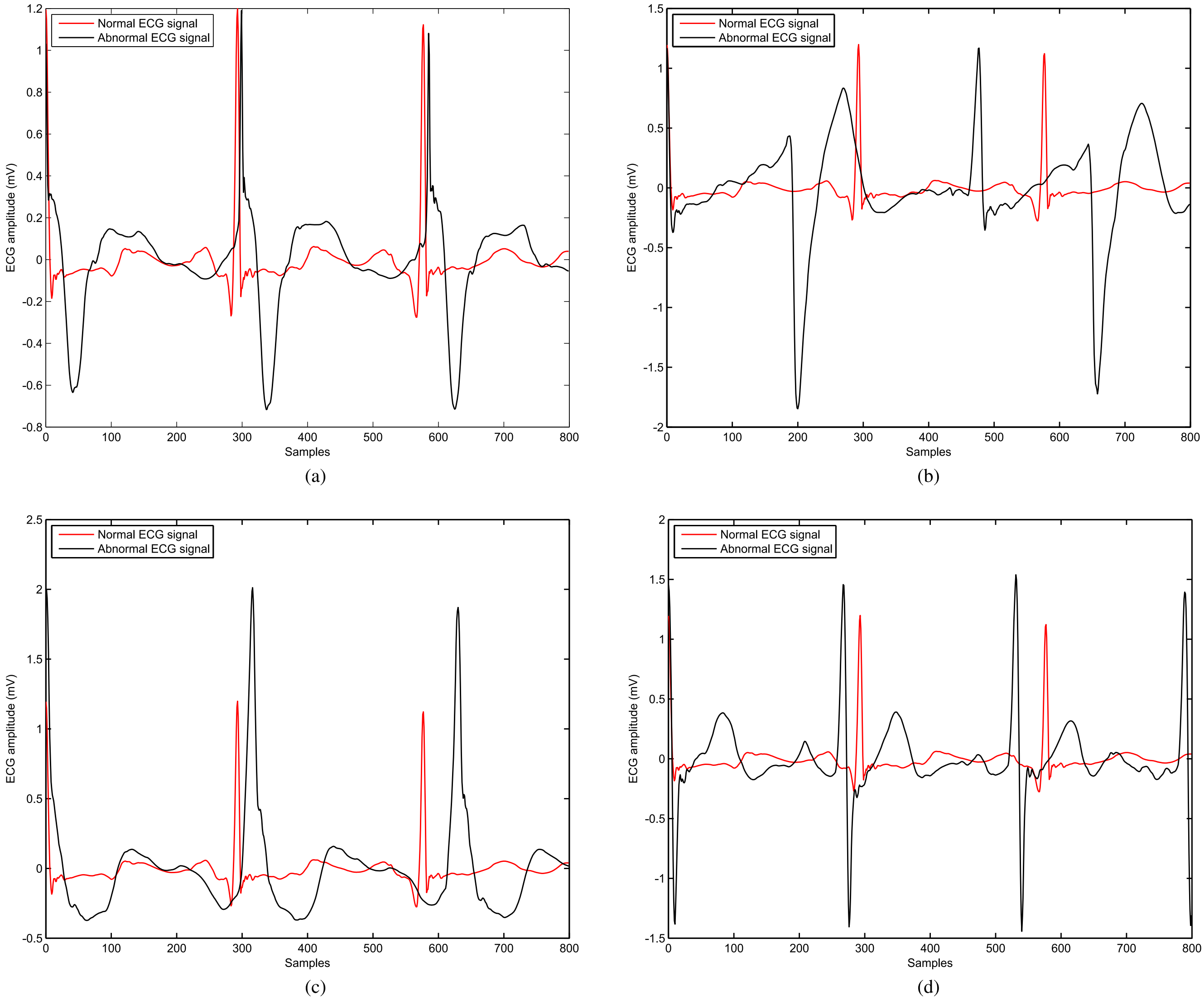


Fig. 8. Normal and abnormal ECG signals: (a) 100 and 104 ECG signals, (b) 100 and 200 ECG signals, (c) 100 and 214 ECG signals, (d) 100 and 230 ECG signals.

et al. (2009) proposed neural networks ECG classification based on type-2 fuzzy clustering and/without wavelet transform, respectively. In Zbay et al. (2011), the main idea was to proceed by three blocs: the type-2 fuzzy clustering system followed by the wavelet transform feature extraction model to finally proceed to the neural network system. Those studies deals with type-2 fuzzy logic and neural networks separately. While in Phong and Thien (2009), and as stated by Melin and Castillo (2013), authors proposed a type-2 TSK fuzzy system for ECG arrhythmic classification. In this study, the generalized bell primary membership function is used to examine the performance of the system with different shapes of membership functions. And in Tan et al. (2007) the feasibility of using a type-2 fuzzy system for ECG arrhythmic beat classification was studied. Authors used a combination of the fuzzy C-means clustering algorithm and the amount of dispersion in each cluster to classify ECG arrhythmic beats: normal sinus rhythm, ventricular fibrillation and ventricular tachycardia. In this paper, we deal with neuro type-2 fuzzy logic systems in which type-2 fuzzy parameters are fitted by using neural networks. Especially in this section, we introduce the BIT2FNN as a new automatic monitoring system for detecting abnormalities in a normal ECG signal. We considered data from the MIT-BIH Physionet arrhythmia database (Goldberger et al., 2000; Moody and Mark, 2001). We used five data-files of one minute length-recording. (one data-files for normal ECG signal 100, and four data-files for abnormal ECG signal defined by 104, 200, 214 and 230 files). These recordings were scanned with 11-bit resolution over a range of 10 mV. Since the mathematical model of an ECG is not defined, the proposed Beta fuzzy neural network is applied for testing the regularity of heartbeats and then can detect if any abnormality occurs.

Before to process with neural network computation, a preprocessing stage of the ECG signal is performed. In this phase a baseline wander removal and a de-noising of the ECG signal were carried out by using multiresolution wavelet transform. The baseline wanders are generally errors that come from respiration and are less than 0.3 Hz. Those errors are removed by first smoothing the original ECG signal by applying the moving average method with a span size equal to 150. Then the smoothed signal is subtracted from the original signal to obtain a free from baseline drift ECG signal. While the de-noise phase, which consists in eliminating different noise structures, was established by using a fourth order Daubechies wavelet. The chosen wavelet transform decomposed the ECG signal on N components. And then an automatic threshold technique is applying to each level to detail coefficients. Finally, the ECG signal is erected based on the approximated original coefficients of level N and modified coefficients of level 1 to N. When the de-noised phase is achieved, R-peaks ECG signal will be detected by using wavelet transform decomposition with high level value. The detection of R-peaks is benefit in our study in the phase of training and testing of neural

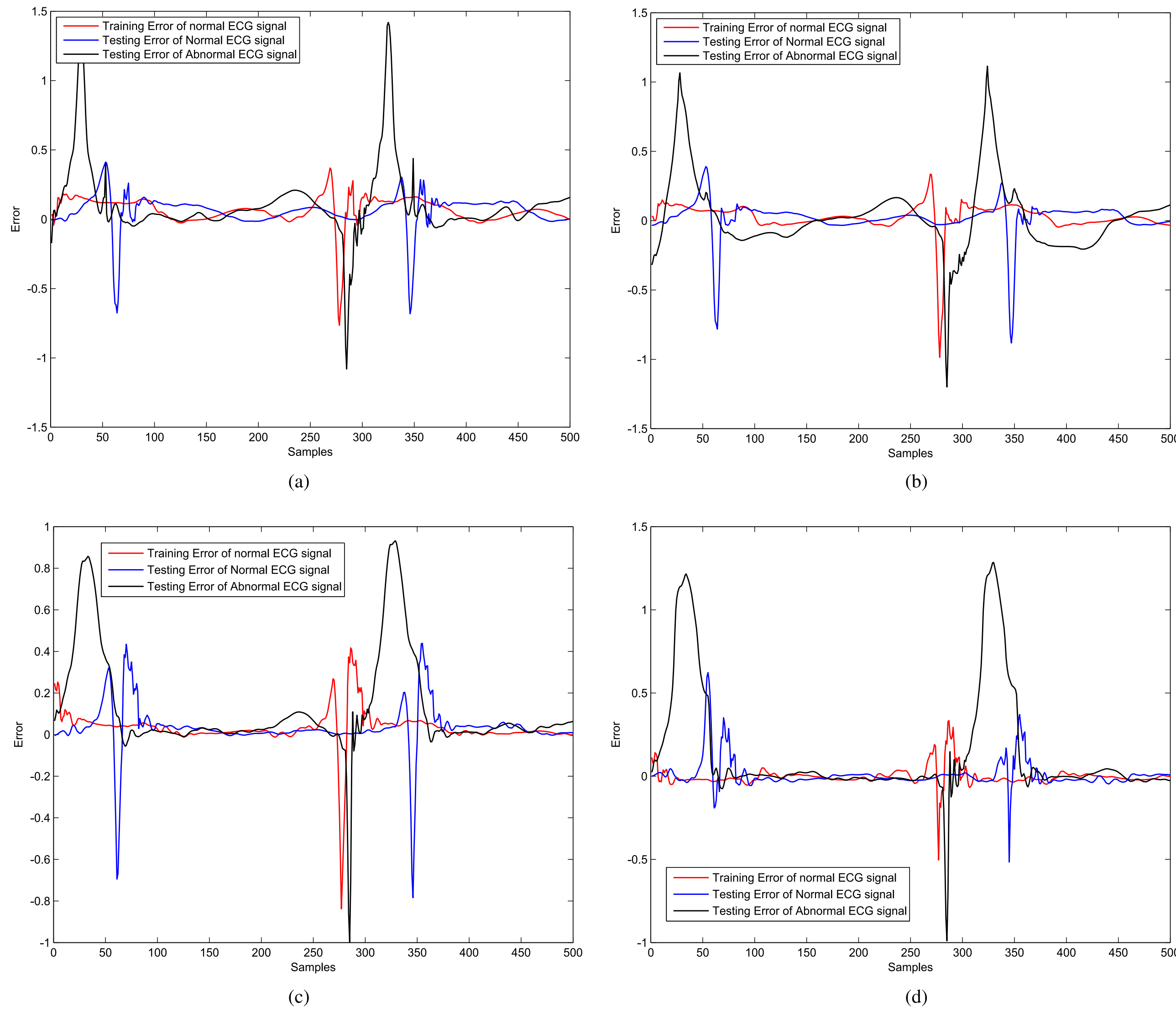


Fig. 9. Results of: Training errors of normal 100-ECG signal, testing errors of normal 100 and abnormal 104-ECG of: (a) GT1FNN, (b) GIT2FNN, (c) BT1FNN, (d) BIT2FNN.

networks. Once the preprocessing stage is realized, the whole resultant signal from the de-noising phase is used to avoid any risk of losing data and to have a better approximation result in the neural network. In all the experimentation part we used only a same one channel recording for all the database. Normal recording data (100-ECG) are divided into two sets for the training and testing phases. While the abnormal data are supposed to be the sudden abnormalities and arrhythmia on the normal ECG. Hence, those abnormal recordings are used for a second testing phase which is corresponding to a check phase of the automatic fuzzy-neural system ability diagnose the arrhythmia. As described earlier the idea is to predict x_{k+1} from neighbor samples $x_{k-n}, n > 0$. We defined the learning data pairs by: $[x(t-15), x(t-10), x(t-5); x(t)]$. We designed the first 500 patterns of an ECG signal s(1001),… s(1500) for the training phase and the remaining 500 patterns are used for testing the output design. We considered the same models of fuzzy neural networks described in the previous section, GT1FNN, GIT2FNN, BT1FNN and BIT2FNN. All the considered systems have three inputs which each is defined by five membership functions, one output and are specified by 125 rules, which corresponds to the maximum number of rules. The main idea is to use 1000 data pairs beginning from the first R peaks of a normal ECG signal. Those data are used for the training and testing phases. Then a 500 data pairs from an abnormal ECG signal and beginning from the first R peaks, are used for checking the abnormality

Table 9
Used normal and abnormal database classes.

ECG class	Database name			
Normal	100			
Abnormal	104	200	214	230

and the sudden change of the ECG signal. Performance is evaluated using the root mean square error (RMSE) as presented in Eq. (44). Then a threshold is applied to the $RMSE$ value to classify the signal output as normal or abnormal. Table 9 shows the normal and abnormal used classes in our study analysis. The classification of those beats as normal or abnormal have been detailed in Goldberger et al. (2000) and Moody and Mark (2001).

Baseline wander removal and de-noising results of the ECG-100 original signal, are depicted in Fig. 7. The same process of the preprocessing treatment (Baseline wander removal and de-noising) is applied to abnormal signals. As follows from Fig. 8(a)–(d) shown next, resultant used normal and abnormal ECG signals are depicted. All signals begin from the first R-peaks to have an equal comparison in the testing phase.

For the training process, as previously taken, we fix the α parameter to 0.2. Regarding the parameter initialization of the BT1FNN, initial

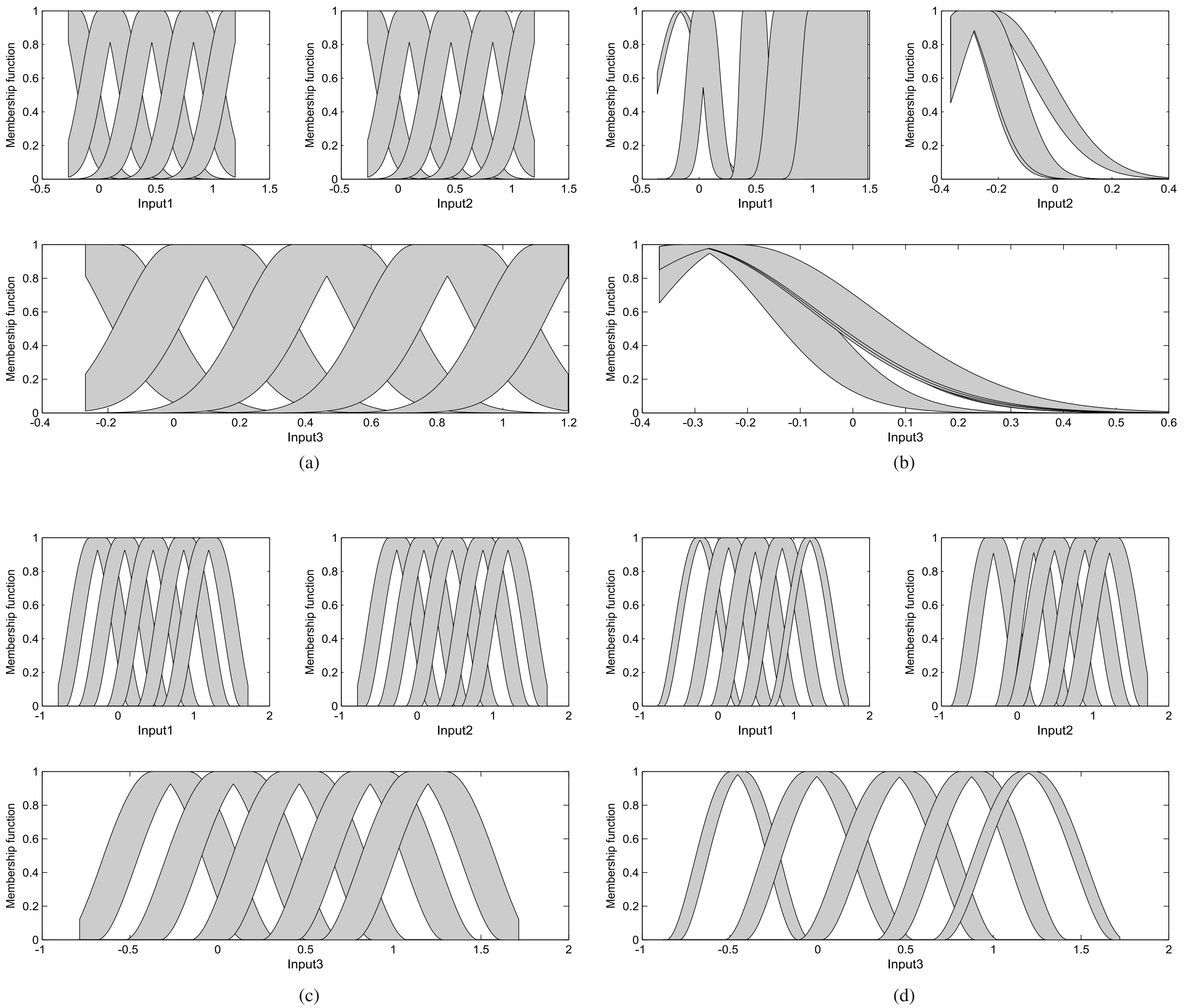


Fig. 10. (a) Initial MFs of GIT2FNN, (b) Final obtained MFs of GIT2FNN, (c) Initial MFs of BIT2FNN, (d) Final obtained MFs of BIT2FNN,.

Table 10
RMSE training and testing errors of normal 100-ECG signal versus abnormal 104/200/214/230-ECG signals.

RMSE of ECG-signals	GT1FNN	GIT2FNN	BT1FNN,	BIT2FNN
Training ECG-100	0.1219	0.1110	0.0992	0.0555
Testing ECG-100	0.1365	0.1311	0.1227	0.0898
Testing ECG-104	0.2857	0.2667	0.2761	0.3745
Testing ECG-200	0.4050	0.4448	0.4969	0.5153
Testing ECG-214	0.4934	0.4982	0.5207	0.5466
Testing ECG-230	0.2872	0.2889	0.2921	0.2986

values of standard deviation σ, p, q and center c were fixed in the good of having equal comparisons. Regarding the BIT2FNN, where we are using beta primary membership functions with uncertain center, we initialized c_1 and c_2 values to respectively $c - 0.1$ and $c + 0.1$. The same strategy was used for gaussian type-1 and type-2 systems. Table 10 summarizes the obtained root mean square errors of the training and testing phases of the normal ECG signal versus abnormal signals. We can remark that the BIT2FNN gives the smallest errors values in both the training and testing stages of normal ECG signal. Whereas in testing abnormal ECG signals, BIT2FNN provides the biggest RMSE values. Hence we can deduce the great ability of this model system, BIT2FNN, in handling more uncertainties, approximating the real signal and identifying abnormal inaccuracies. The results exhibit that within same conditions, the BIT2FNN gives clearly better performance regarding other systems. In this case study, an RMSE threshold for detecting fatal abnormalities can be fixed to a value of 0.2.

For visual representation of those errors, we depicted the instantaneous errors results (Training and testing errors of normal ECG signal, Testing error of Abnormal ECG signal) over 500 samples, for the 100 and 104 ECG signals, in the following Fig. 9(a), (b), (c) and (d). As can be seen the BIT2FNN confirmed the previous results while it represents the biggest values of testing error of abnormal ECG signals. An important implication of these findings is that the BIT2FNN can detect potentially fatal ECG abnormalities.

Initial and final obtained membership functions for GT1FNN and BIT2FNN systems are depicted in Fig. 10.

The experimental results exhibit that type-2 beta neural networks are able to predict relations in non linear data science equations even in the presence of different levels of noise. The use of Beta membership function which has additional degree of freedom helps the fuzzy network to achieve a better behavior.

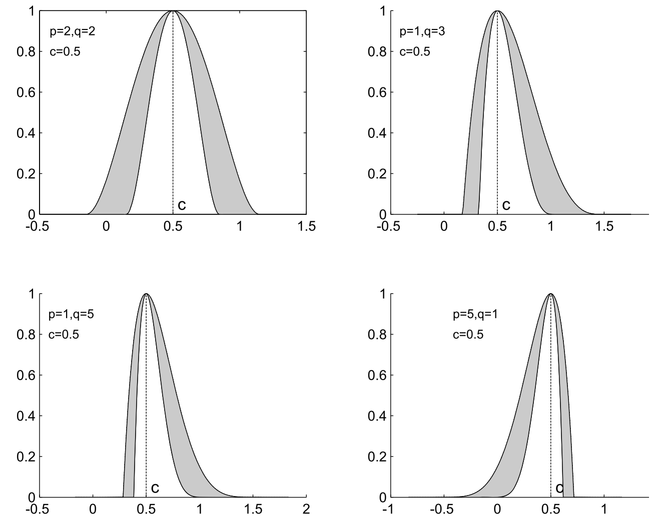


Fig. 11. FOU for Beta primary MF with uncertain standard deviation.

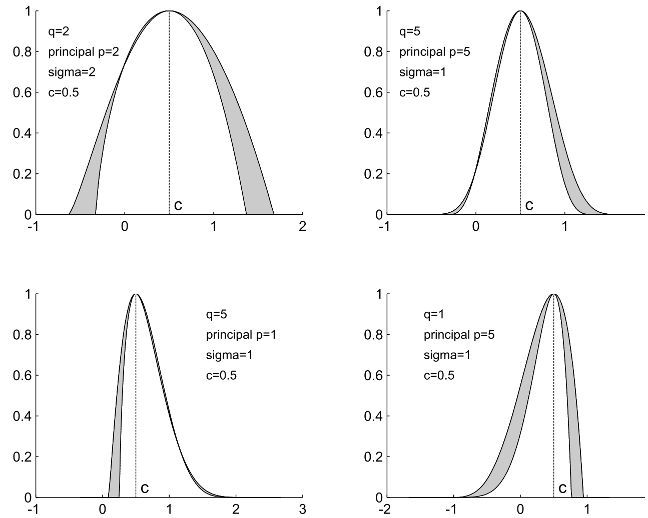


Fig. 12. A Beta primary MF with uncertain: p.

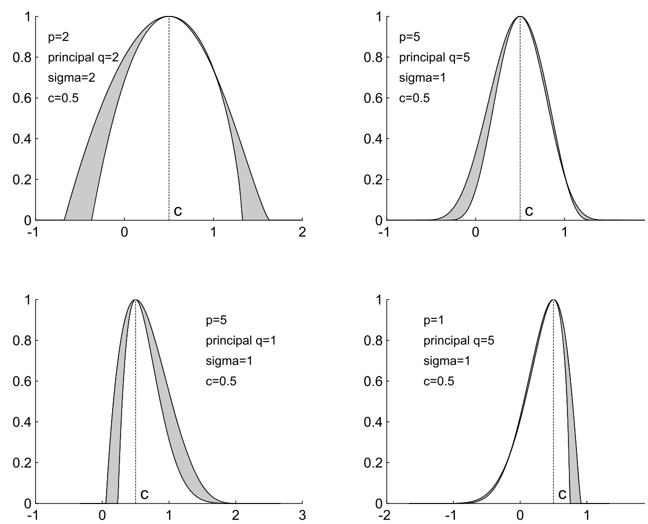


Fig. 13. A Beta primary MF with uncertain: q.

7. Conclusion

In this paper, a Beta Basis Function Interval Type-2 Fuzzy Neural Network BIT2FNN was proposed for real time series applications. Throughout this fuzzy network a type-2 beta fuzzy set was utilized. First order derivatives of type-1 and type-2 beta functions were developed and tuning parameters calculations were ensured based on a given input–output data pairs and the backpropagation learning algorithm. Simulation results have been performed by exploring two examples of time series applications: the Mackay Glass Chaotic Time-Series prediction problem with different setting of parameters and levels of noise and the ECG heart-rate Time Series monitoring problem. The results thus obtained in both examples deduct the great ability of the proposed architecture, BIT2FNN, in handling more uncertainties. They have achieving better forecasting performance and identifying abnormal inaccuracies. The results exhibit that within same conditions, the BIT2FNN have the faculty to deal with uncertain noisy data, and proved to have good approximation ability. Then beta functions may be considered as a supplement and an asset for type-2 fuzzy systems. To yield better total performance, a dynamic learning rate optimization process within an evolving network may be added to the BIT2FNN in further research.

Acknowledgment

The research leading to these results has received funding from the Ministry of Higher Education and Scientific Research of Tunisia under the grant agreement number LR11ES48. Ajith Abraham acknowledges the support from the framework of the IT4Innovations Center of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Program 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

Annexe.

Beta primary membership function with uncertain standard deviation σ. In this case, a beta primary membership function an uncertain σ, $\sigma \in [\sigma_1, \sigma_2]$, and fixed p, q, c, is expressed as follows:

$$\beta(x) = \left(1 + \frac{(p+q)(x-c)}{\sigma p}\right)^p \left(1 - \frac{(p+q)(c-x)}{\sigma q}\right)^q \quad (45)$$
$$with\ \sigma \in [\sigma_1, \sigma_2].$$

An example of this case is illustrated in Fig. 11.

Similarly, upper and lower membership functions can be expressed by the following relations:

$$\bar{\mu}_{\tilde{A}}(x) = \beta(x; c, \sigma_2, p, q) \quad (46)$$

$$\underline{\mu}_{\tilde{A}}(x) = \beta(x; c, \sigma_1, p, q). \quad (47)$$

Beta primary MF with uncertain p. A beta primary membership function with an uncertain p value, $p \in [p_1, p_2]$, is defined by the following equation in which to each value of p corresponds a different MF. An illustrative example of this type-2 fuzzy set is given in Fig. 12.

$$\beta(x) = \left(1 + \frac{(p+q)(x-c)}{\sigma p}\right)^p \left(1 - \frac{(p+q)(c-x)}{\sigma q}\right)^q \quad (48)$$
$$p \in [p_1, p_2].$$

Expressions of the associated UMF and LMF to this case function are given by Eqs. (49) and (50), respectively:

$$\bar{\mu}_{\tilde{A}}(x) = \begin{cases} \max(\beta(x; c, \sigma, p_1, q), \beta(x; c, \sigma, p_2, q)) \\ \forall x \in X \end{cases} \quad (49)$$

$$\underline{\mu}_{\tilde{A}}(x) = \begin{cases} \max(\beta(x; c, \sigma, p_1, q), \beta(x; c, \sigma, p_2, q)) \\ \forall x \in X. \end{cases} \quad (50)$$

Beta primary MF with uncertain q. A beta primary membership function with an uncertain q value, $q \in [q_1, q_2]$ is defined by the following equation, in which to each value of q, corresponds a different MF. An example type of this type-2 fuzzy set can be illustrated in Fig. 13.

$$\beta(x) = \left(1 + \frac{(p+q)(x-c)}{\sigma p}\right)^p \left(1 - \frac{(p+q)(c-x)}{\sigma q}\right)^q \quad (51)$$
$$with\ q \in [q_1, q_2].$$

For this type-2 fuzzy set, the UMF and LMF can be defined by Eqs. (52) and (53), respectively:

$$\bar{\mu}_{\tilde{A}}(x) = \begin{cases} \max(\beta(x; c, \sigma, p, q_1), \beta(x; c, \sigma, p, q_2)) \\ \forall x \in X \end{cases} \quad (52)$$

$$\underline{\mu}_{\tilde{A}}(x) = \begin{cases} \min(\beta(x; c, \sigma, p, q_1), \beta(x; c, \sigma, p, q_2)) \\ \forall x \in X. \end{cases} \quad (53)$$

References

Alimi, M., 1997a. The beta fuzzy system: approximation of standard membership functions. In: Proceedings 17 Eme Journees Tunisiennes D'Electrotechnique Et D'Automatique, JTEA'97, Vol. 1, Tunisia, pp. 108–112.

Alimi, M., 1997b. The recognition of Arabic handwritten characters With the Beta neuro Fuzzy network. In: Proceedings 17 Eme Journees Tunisiennes D'Electrotechnique Et D'Automatique, JTEA'97, Vol. 1, Tunisia, pp. 349–356.

Alimi, A.M., 2000. The Beta system : toward a change in our use of neuro-fuzzy systems. Int. J. Manag. 15–19. Invited paper.

Alimi, A.M., 2003. Beta neuro-fuzzy systems. TASK Quart. J. Spec. Issue Neural Netw. 7 (1), 23–41.

Alimi, A.M., Hassine, R., Selmi, M., 2000. Beta fuzzy logic systems: Approximation properties in the siso case. Int. J. Appl. Math. Comput. Sci. 10 (4), 857–875.

Alimi, A.M., Hassine, R., Selmi, M., 2003. Beta fuzzy logic systems: Aproximation propreties in the mimo case. Int. J. Appl. Math. Comput. Sci. 13 (2), 101–114.

Baklouti, N., Alimi, A.M., Abraham, A., 2015. Interval type-2 beta fuzzy basis functions: Some properties and their first-order derivatives. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC, pp. 2227–2232. http://dx.doi.org/10.1109/SMC.2015.389.

Bezine, H., Alimi, A.M., Derbel, N., 2003. Handwriting trajectory movements controlled by a Bêta-elliptic model. In: 7th International Conference on Document Analysis and Recognition, ICDAR, 2-Volume Set, 3-6 August 2003, Scotland, UK, pp. 1228–1232.

Bezine, H., Kefi, M., Alimi, A.M., 2007. On the beta-elliptic model for the control of the human arm movement. IJPRAI 21 (1), 5–19.

Bouaziz, S., Alimi, A.M., Abraham, A., 2014. Pso-based update memory for improved harmony search algorithm to the evolution of fbbfnt' parameters. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014, pp. 1951–1958.

Bouaziz, S., Dhahri, H., Alimi, A.M., Abraham, A., 2013. A hybrid learning algorithm for evolving flexible beta basis function neural tree model. Neurocomputing 117, 107–117.

Castro, J.R., Castillo, O., Melin, P., Rodriguez-Diaz, A., 2009. A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. Inform. Sci. 179 (13), 2175–2193. http://dx.doi.org/10.1016/j.ins.2008.10.016.

Ceylan, R., Zbay, Y., Karlik, B., 2009. A novel approach for classification of {ECG} arrhythmias: Type-2 fuzzy clustering neural network. Expert Syst. Appl. 36 (3, Part 2), 6721–6726. http://dx.doi.org/10.1016/j.eswa.2008.08.028. URL http://www.sciencedirect.com/science/article/pii/S0957417408005745.

Chua, T.W., Tan, W.W., 2011. Non-singleton genetic fuzzy logic system for arrhythmias classification. Eng. Appl. Artif. Intell. 24 (2), 251–259. http://dx.doi.org/10.1016/j.engappai.2010.10.003. URL http://www.sciencedirect.com/science/article/pii/S0952197610001892.

Das, A.K., Subramanian, K., Sundaram, S., 2015. An evolving interval type-2 neurofuzzy inference system and its metacognitive sequential learning algorithm. IEEE Trans. Fuzzy Syst. 23 (6), 2080–2093. http://dx.doi.org/10.1109/TFUZZ.2015.2403793.

de Chazal, P., Celler, B.G., Reilly, R.B., 2000. Using wavelet coefficients for the classification of the electrocardiogram. In: Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Cat. No.00CH37143), Vol. 1, pp. 64–67. http://dx.doi.org/10.1109/IEMBS.2000.900669.

Fletcher, R., Reeves, C.M., 1964. Function minimization by conjugate gradients. Comput. J. 7 (2), 149–154. http://dx.doi.org/10.1093/comjnl/7.2.149.

Gaxiola, F., Melin, P., Valdez, F., 2014a. Recent Advances on Hybrid Approaches for Designing Intelligent Systems. Springer International Publishing, Cham, pp. 51–65. http://dx.doi.org/10.1007/978-3-319-05170-3_4. (Chapter). Neural Network with Fuzzy Weights Using Type-1 and Type-2 Fuzzy Learning with Gaussian Membership Functions.

Gaxiola, F., Melin, P., Valdez, F., Castillo, O., 2014b. Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction. Inform. Sci. 260, 1–14. http://dx.doi.org/10.1016/j.ins.2013.11.006. URL http://www.sciencedirect.com/science/article/pii/S0020025513007937.

Gaxiola, F., Melin, P., Valdez, F., Castillo, O., 2015. Generalized type-2 fuzzy weight adjustment for backpropagation neural networks in time series prediction. Inform. Sci. 325, 159–174. http://dx.doi.org/10.1016/j.ins.2015.07.020. URL http://www.sciencedirect.com/science/article/pii/S0020025515005125.

Gler, I., Beyli, E.D., 2004. Application of adaptive neuro-fuzzy inference system for detection of electrocardiographic changes in patients with partial epilepsy using feature extraction. Expert Syst. Appl. 27 (3), 323–330. http://dx.doi.org/10.1016/j.eswa.2004.05.001. URL http://www.sciencedirect.com/science/article/pii/S0957417404000363.

Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K., Stanley, H.E., 2000. PhysioBank, PhysioToolkit, and PhysioNet. Circulation 101 (23), e215–e220. http://dx.doi.org/10.1161/01.CIR.101.23.e215. URL http://circ.ahajournals.org/content/101/23/e215. arXiv:http://circ.ahajournals.org/content/101/23/e215.full.pdf.

Ishibuchi, H., Morioka, K., Turksen, I., 1995. Learning by fuzzified neural networks. Internat. J. Approx. Reason. 13 (4), 327–358. http://dx.doi.org/10.1016/0888-613X(95)00060-T. URL http://www.sciencedirect.com/science/article/pii/0888613X9500060T.

Ishibuchi, H., Tanaka, H., Okada, H., 1993. Fuzzy neural networks with fuzzy weights and fuzzy biases. In: IEEE International Conference on Neural Networks, 1993, Vol.3, pp. 1650–1655. http://dx.doi.org/10.1109/ICNN.1993.298804.

Jang, J.S.R., 1993. Anfis: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. 23 (3), 665–685. http://dx.doi.org/10.1109/21.256541.

Juang, C.F., Huang, R.B., Cheng, W.Y., 2010. An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems. IEEE Trans. Fuzzy Syst. 18 (4), 686–699. http://dx.doi.org/10.1109/TFUZZ.2010.2046904.

Juang, C.-F., Lin, C.-T., 1998. An online self-constructing neural fuzzy inference network and its applications. IEEE Trans. Fuzzy Syst. 6 (1), 12–32. http://dx.doi.org/10.1109/91.660805.

Juang, C.-F., Tsao, Y.-W., 2008. A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning. IEEE Trans. Fuzzy Syst. 16 (6), 1411–1424. http://dx.doi.org/10.1109/TFUZZ.2008.925907.

Karnik, N.N., Mendel, J.M., 1999. Applications of Type-2 fuzzy logic systems to forecasting of time-series. Inform. Sci. 120 (1–4), 89–111.

Lee, C.-H., Lin, Y.-C., Lai, W.-Y., 2003. Systems identification using type-2 fuzzy neural network (type-2 fnn) systems. In: 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2003. Proceedings. Vol. 3, pp. 1264–1269. http://dx.doi.org/10.1109/CIRA.2003.1222178.

Liang, Q., Mendel, J., 2000. Interval type-2 fuzzy logic systems: theory and design. IEEE Trans. Fuzzy Syst. 8 (5), 535–550. http://dx.doi.org/10.1109/91.873577.

Ltaief, M., Bezine, H., Alimi, A.M., 2012. A neuro-beta-elliptic model for handwriting generation movements. In: ICFHR, pp. 803–808. http://dx.doi.org/10.1109/ICFHR.2012.161.

MaNdez, G.M., De Los Angeles Hernández, M., 2013. Hybrid learning mechanism for interval a2-c1 type-2 non-singleton type-2 takagi-sugeno-kang fuzzy logic systems. Inform. Sci. 220, 149–169. http://dx.doi.org/10.1016/j.ins.2012.01.024.

Melin, P., Castillo, O., 2013. A review on the applications of type-2 fuzzy logic in classification and pattern recognition. Expert Syst. Appl. 40 (13), 5413–5423. http://dx.doi.org/10.1016/j.eswa.2013.03.020. URL http://www.sciencedirect.com/science/article/pii/S0957417413001826.

Mendel, J., 2001a. Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall PTR.

Mendel, J., 2001b. Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River NJ.

Mendel, J., 2004. Computing derivatives in interval type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. 12 (1), 84–98. http://dx.doi.org/10.1109/TFUZZ.2003.822681.

Mendel, J., John, R., 2002. Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. 10 (2), 117–127. http://dx.doi.org/10.1109/91.995115.

Mendel, J., Mouzouris, G., 1997. Designing fuzzy logic systems. IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process. 44 (11), 885–895. http://dx.doi.org/10.1109/82.644042.

Mjahad, A., Rosado-Muoz, A., Bataller-Mompen, M., Francs-Vllora, J., Guerrero-Martnez, J., 2017. Ventricular fibrillation and tachycardia detection from surface {ECG} using time-frequency representation images as input dataset for machine learning. Comput. Methods Programs Biomed. 141, 119–127. http://dx.doi.org/10.1016/j.cmpb.2017.02.010. URL http://www.sciencedirect.com/science/article/pii/S0169260716304825.

Moody, G.B., Mark, R.G., 2001. The impact of the mit-bih arrhythmia database. IEEE Eng. Med. Biol. Mag. 20 (3), 45–50. http://dx.doi.org/10.1109/51.932724.

Mndez, G.M., de los Angeles Hernandez, M., 2009. Hybrid learning for interval type-2 fuzzy logic systems based on orthogonal least-squares and back-propagation methods. Inform. Sci. 179 (13), 2146–2157. http://dx.doi.org/10.1016/j.ins.2008.08.

008. URL http://www.sciencedirect.com/science/article/pii/S002002550800337X, Special Section on High Order Fuzzy Sets.

Olivas, F., Valdez, F., Castillo, O., 2014. Recent Advances on Hybrid Approaches for Designing Intelligent Systems. Springer International Publishing, Cham, pp. 67–78. http://dx.doi.org/10.1007/978-3-319-05170-3_5. (Chapter). A Comparative Study of Membership Functions for an Interval Type-2 Fuzzy System used to Dynamic Parameter Adaptation in Particle Swarm Optimization.

Phong, P.A., Thien, K.Q., 2009. Classification of cardiac arrhythmias using interval type-2 tsk fuzzy system. In: 2009 International Conference on Knowledge and Systems Engineering, pp. 1–6. http://dx.doi.org/10.1109/KSE.2009.19.

Rai, H.M., Trivedi, A., Shukla, S., 2013. {ECG} signal processing for abnormalities detection using multi-resolution wavelet transform and Artificial Neural Network classifier. Measurement 46 (9), 3238–3246. http://dx.doi.org/10.1016/j.measurement.2013.05.021. URL http://www.sciencedirect.com/science/article/pii/S0263224113002133.

Ranganathan, G., Rangarajan, R., Bindhu, V., 2012. Estimation of heart rate signals for mental stress assessment using neuro fuzzy technique. Appl. Soft Comput. 12 (8), 1978–1984. http://dx.doi.org/10.1016/j.asoc.2012.03.019. URL http://www.sciencedirect.com/science/article/pii/S1568494612001093.

Rubio-Solis, A., Panoutsos, G., 2015. Interval type-2 radial basis function neural network: A modeling framework. IEEE Trans. Fuzzy Syst. 23 (2), 457–473. http://dx.doi.org/10.1109/TFUZZ.2014.2315656.

Seena, V., Yomas, J., 2014. A review on feature extraction and denoising of ecg signal using wavelet transform. In: 2014 2nd International Conference on Devices, Circuits and Systems, ICDCS, pp. 1–6. http://dx.doi.org/10.1109/ICDCSyst.2014.6926190.

Su, L., Zhao, G., (2005) 2005. De-noising of ecg signal using translation- invariant wavelet de-noising method with improved thresholding. In: 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, pp. 5946–5949. http://dx.doi.org/10.1109/IEMBS.2005.1615845.

Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. 15 (1), 116–132.

Tan, W.W., Foo, C.L., Chua, T.W., 2007. Type-2 fuzzy system for ECG arrhythmic classification. In: 2007 IEEE International Fuzzy Systems Conference, pp. 1–6. http://dx.doi.org/10.1109/FUZZY.2007.4295478.

Tung, S., Quek, C., Guan, C., 2013. et2fis: An evolving type-2 neural fuzzy inference system. Inform. Sci. 220, 124–148. http://dx.doi.org/10.1016/j.ins.2012.02.031. URL http://www.sciencedirect.com/science/article/pii/S0020025512001491, Online Fuzzy Machine Learning and Data Mining.

Uncu, O., Turksen, I., 2007. Discrete interval type 2 fuzzy system models using uncertainty in learning parameters. IEEE Trans. Fuzzy Syst. 15 (1), 90–106. http://dx.doi.org/10.1109/TFUZZ.2006.889765.

Wang, L.X., 1992. Fuzzy systems are universal approximators. In: IEEE International Conference on Fuzzy Systems, 1992, pp. 1163–1170. http://dx.doi.org/10.1109/FUZZY.1992.258721.

Wang, C., Cheng, C., Lee, T., 2004. Dynamical optimal training for interval type-2 fuzzy neural network (t2fnn). IEEE Trans. Syst. Man Cybern. B 34 (3), 1462–1477. http://dx.doi.org/10.1109/TSMCB.2004.825927.

Wang, L.X., Mendel, J.M., 1992. Back-propagation fuzzy system as nonlinear dynamic system identifiers. In: IEEE International Conference on Fuzzy Systems, 1992, pp. 1409–1418. http://dx.doi.org/10.1109/FUZZY.1992.258711.

Wu, S., Er, M.J., 2000. Dynamic fuzzy neural networks-a novel approach to function approximation. IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) 30 (2), 358–364. http://dx.doi.org/10.1109/3477.836384.

Xiong, P., Wang, H., Liu, M., Zhou, S., Hou, Z., Liu, X., 2016. {ECG} signal enhancement based on improved denoising auto-encoder. Eng. Appl. Artif. Intell. 52, 194–202. http://dx.doi.org/10.1016/j.engappai.2016.02.015. URL http://www.sciencedirect.com/science/article/pii/S0952197616300331.

Zarandi, M.F., Torshizi, A.D., Turksen, I., Rezaee, B., 2013. A new indirect approach to the type-2 fuzzy systems modeling and design. Inform. Sci. 232, 346–365. http://dx.doi.org/10.1016/j.ins.2012.12.017. URL http://www.sciencedirect.com/science/article/pii/S002002551200802X.

Zbay, Y., Ceylan, R., Karlik, B., 2011. Integration of type-2 fuzzy clustering and wavelet transform in a neural network based {ECG} classifier. Expert Syst. Appl. 38 (1), 1004–1010. http://dx.doi.org/10.1016/j.eswa.2010.07.118. URL http://www.sciencedirect.com/science/article/pii/S0957417410007463.