Proceedings Paper:
Jafari, R orcid.org/0000-0001-7298-2363, Razvarz, S, Gegov, A et al. (1 more author) (2018) Fuzzy Modeling for Uncertain Nonlinear Systems Using Fuzzy Equations and Z-Numbers. In: Advances in Computational Intelligence Systems. UKCI-2018, 05-07 Sep 2018, Nottingham, UK. Springer Verlag , pp. 96-107. ISBN 9783319979816

https://doi.org/10.1007/978-3-319-97982-3_8

Fuzzy Modeling for Uncertain Nonlinear Systems using Fuzzy Equations and Z-numbers

Raheleh Jafari[1[0000-0001-7298-2363]], Sina Razvarz[2] ,Alexander Gegov[3] and satyam paul[4]

[1] Center for Artificial Intelligence Research (CAIR), University of Agder, 4879 Grimstad, Norway
[2] Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), 07360 Mexico City, Mexico;
[3] School of Computing, University of Portsmouth, Buckingham Building, Portsmouth PO13HE, UK
[4] School of Engineering and Sciences, Tecnológico de Monterrey, Monterrey, Nuevo Leon, Mexico, 64849.
jafari3339@yahoo.com

Abstract. In this paper, the uncertainty property is represented by Z-number as the coefficients and variables of the fuzzy equation. This modification for the fuzzy equation is suitable for nonlinear system modeling with uncertain parameters. Here, we use fuzzy equations as the models for the uncertain nonlinear systems. The modeling of the uncertain nonlinear systems is to find the coefficients of the fuzzy equation. However, it is very difficult to obtain Z-number coefficients of the fuzzy equations.

Taking into consideration the modeling case at par with uncertain nonlinear systems, the implementation of neural network technique is contributed in the complex way of dealing the appropriate coefficients of the fuzzy equations. We use the neural network method to approximate Z-number coefficients of the fuzzy equations.

Keywords: Fuzzy Modeling, Z-number, Uncertain Nonlinear System.

1 Introduction

An exceptional case of uncertain system modeling at par with fuzzy equation is fuzzy polynomial interpolation. Polynomials have been used with fuzzy coefficients in order to interpolate uncertain data that are expressed using fuzzy numbers [1]. Interpolation methodology has been broadly utilized for function approximation as well as system identification [2]. In [3], the fuzzy polynomial interpolation is applied for system modeling. The theory problem associated with polynomial interpolation is researched in [4]. It elaborates that the interpolation of the function includes time complexity at par with data points. In [5], two-dimensional polynomial interpolation is demonstrated. Smooth function approximation has been broadly implemented currently [6]. It yields a model by utilizing Lagrange interpolating polynomials at the points of product grids [1,7].

However if it involves uncertainties in the interpolation points, the above suggested techniques will not work appropriately.

The fuzzy equation can be regarded as a generalized form of the fuzzy polynomial. Compared with the normal fuzzy systems, the fuzzy equations are more easy to be applied. There are several approaches to construct the fuzzy equations. [8] used the fuzzy number on parametric shapes and replaced the original fuzzy equations with crisp linear systems. [9] proposed the homotypic analysis technique. [10] used the Newton methodology. In [11] the solution associated to the fuzzy equations is studied by the fixed point technique. The numerical solution associated to the fuzzy equations can be extracted by iterative technique [12], interpolation technique [13] and the Runge-Kutta technique [14]. The neural networks may also be used to solve fuzzy equations. In [15], the simple fuzzy quadratic equation is resolved by the neural network method. [16] extended the result of [15] to fuzzy polynomial equations. In [17], the solution of dual fuzzy equation is obtained by neural networks. A matrix pattern associated with the neural learning has been quoted in [18]. The predictor-corrector approach is applied in [19].

The decisions are carried out based on knowledge. In order to make the decision fruitful, the knowledge acquired must be credible. Z-numbers connect to the reliability of knowledge [20]. Many fields related to the analysis of the decisions use the ideas of Z-numbers. Z-numbers are much less complex to calculate when compared to nonlinear system modeling methods. Z-number is abundantly adequate number than the fuzzy number. Although Z-numbers are implemented in many literatures, from theoretical point of view this approach is not certified completely. There are few structure based on the theoretical concept of Z-numbers [21]. [22] gave an inception which results in the extension of Z-numbers. [23] proposed a theorem to transfer Z-numbers to the usual fuzzy sets. In [20] a novel approach was followed for the conversion of Z-number into fuzzy number.

In this paper, we use fuzzy equations to model the uncertain nonlinear systems, where the coefficients and variables are Z-numbers. Z-number is a novel idea that is subjected to a higher potential in order to illustrate the information of the human being as well as to use in information processing [20]. Z-numbers can be regarded as to answer questions and carry out the decisions [24]. This paper is one of the first attempts in finding the coefficients of fuzzy equations based on Z-numbers. We use the neural network method to approximate the coefficients of the fuzzy equations. The standard backpropagation method is modified, such that Z-numbers in the fuzzy equations can be trained.

2 Nonlinear system modeling with fuzzy equations and Z-numbers

A general discrete-time nonlinear system can be described as

$$\overline{x}_{k+1} = \overline{p}[\overline{x}_k, w_k], s_k = \overline{q}[\overline{x}_k] \quad (1)$$

Here we consider $W_k \in \Re^u$ as the input vector, $\overline{x}_k \in \Re^l$ is regarded as an internal state vector and $S_k \in \Re^m$ is the output vector. $\overline{p}$ and $\overline{q}$ are noted as generalized nonlinear smooth functions $\overline{p}, \overline{q} \in C^\infty$. Define $S_k = [s_{k+1}^T, s_k^T, \ldots]^T$ and $W_k = [W_{k+1}^T, W_k^T, \ldots]^T$.

Suppose$\frac{\partial S}{\partial \overline{x}}$ is non-singular at the instance $\overline{x}_k = 0$, $W_k = 0$, this will create a path towards the following model

$$s_k = \Omega[s_{k-1}^T, s_{k-2}^T, \cdots w_k^T, w_{k-1}^T, \cdots] \tag{2}$$

Where $\Omega(\cdot)$ is an nonlinear difference equation exhibiting the plant dynamics, W_k and S_k are computable scalar input and output respectively. The nonlinear system which is represented by (2) is implied as a NARMA model. The input of the system with incorporated nonlinearity is considered to be as

$$x_k = [s_{k-1}^T, s_{k-2}^T, \cdots w_k^T, w_{k-1}^T, \cdots]^T$$

Taking into consideration the nonlinear systems as mentioned in (plant), it can be simplified as the following linear-in-parameter model

$$s_k = \sum_{i=1}^{n} \sum_{j=1}^{m} b_{ij} p_i(x_k) q_j(y_k) \tag{3}$$

here b_{ij} is considered to be the linear parameter, $p_i(x_k)$ and $q_j(y_k)$ are nonlinear functions. The variables related to these functions are quantifying input and output.
The modeling of uncertain nonlinear systems can be achieved by utilizing the linear-in-parameter models linked to fuzzy parameters. We assume the model of the nonlinear systems (3) has uncertainties in the b_{ij} ,x_k and y_k . These uncertainties are in the sense of Z-numbers [25].

Definition 1. If v is: 1) normal, there exists $\vartheta_0 \in \Re$ in such a manner $v(\vartheta_0) = 1$, 2) convex, $v(\gamma\vartheta + (1-\gamma)\vartheta) \geq \min\{v(\vartheta, v(\theta))\}, \forall\vartheta, \theta \in \Re, \forall\gamma \in [0,1]$, 3) upper semi-continuous on $\Re, v(\vartheta) \leq v(\vartheta_0) + \varepsilon, \forall\vartheta \in N(\vartheta_0), \forall\vartheta_0 \in \Re, \forall\varepsilon > 0, N(\vartheta_0)$is a neighborhood, 4) $v^+ = \{\vartheta \in \Re, v(\vartheta) > 0\}$ is compact, then v is a fuzzy variable, $v \in E: R \to [0,1]$.

The fuzzy variable v can be also represented as

$$v = (\underline{v}, \overline{v}) \tag{4}$$

Where $\underline{v}$ is the lower-bound variable and $\overline{v}$ is the upper-bound variable.

Definition 2. A Z -number has two components$Z = [v(\vartheta), p]$. The primary component $v(\vartheta)$ is termed as a restriction on a real-valued uncertain variable ϑ. The secondary component p is a measure of reliability of v . p can be reliability, strength of belief, probability or possibility. When $v(\vartheta)$ is a fuzzy number and p is the probability distribution of , Z -number is defined as Z^+ -number. When both $v(\vartheta)$ and p are fuzzy numbers, Z -number is defined as Z^- -number.

Z^+ -number carries more information than Z^- -number. In this paper, we use the definition of Z^+-number, i.e.,$Z = [v, p], v$ is a fuzzy number and p is a probability distribution.

We use so called membership functions to express the fuzzy number. One of the most popular membership function is the triangular function

$$\mu_v = G(a,b,c) = \begin{cases} \frac{\vartheta - a}{b-a} & a \leq \vartheta \leq b \\ \frac{c-\vartheta}{c-b} & b \leq \vartheta \leq c \end{cases} \; otherwise \; \mu_v = 0 \tag{5}$$

and trapezoidal function

$$\mu_V = G(a,b,c,d) = \begin{cases} \frac{\vartheta-a}{b-a} & a \le \vartheta \le b \\ \frac{d-\vartheta}{d-c} & c \le \vartheta \le d \quad otherwise\ \mu_V = 0 \\ 1 & b \le \vartheta \le c \end{cases} \tag{6}$$

The probability measure is expressed as

$$p = \int_R \mu_v(\vartheta)p(\vartheta)d\vartheta \tag{7}$$

where p is the probability density of ϑ and R is the restriction on p. For discrete Z -numbers

$$p(v) = \sum_{i=1}^{n} \mu_v(\vartheta_i)p(\vartheta_i) \tag{8}$$

Definition 3. The fuzzy number v in association to the α -level is illustrated as

$$[v]^\alpha = \{a \in \Re: v(a) \ge a)\} \tag{9}$$

Where $0 < \alpha \le 1, v \in E$.

Therefore $[v]^0 = v^+ = \{\vartheta \in \Re, v(\vartheta) > 0\}$ Since $\alpha \in [0,1], [v]^\alpha$ is a bounded mentioned as $\underline{v}^\alpha \le [v]^\alpha \le \overline{v}^\alpha$ The α -level of v in midst of $\underline{v}^\alpha$ and $\overline{v}^\alpha$ is explained as

$$[v]^\alpha = (\underline{v}^\alpha, \overline{v}^\alpha)$$

$\underline{v}^\alpha$ and $\overline{v}^\alpha$ signify the function of α. We state $\underline{v}^\alpha = d_A(\alpha), \overline{v}^\alpha d_B(\alpha), \alpha \in [0,1]$.

Definition 4. The α -level of Z -number $Z = (v, P)$ is demonstrated as

$$[Z]^\alpha = ([V]^\alpha, [p]^\alpha) \tag{10}$$

where $0 < \alpha \le 1$. $[p]^\alpha$ is calculated by the Nguyen's theorem

$$[p]^\alpha = p([v]^\alpha) = p([\underline{v}^\alpha, \bar{v}^\alpha]) = [\underline{P}^\alpha, \overline{P}^\alpha]$$

where $p([v]^\alpha) = \{p(\vartheta) | \vartheta \in [v]^\alpha\}$. So $[Z]^\alpha$can be expressed as the form α -level of a fuzzy number

$$[Z]^\alpha = \left(\underline{Z}^\alpha, \overline{Z}^\alpha\right) = ((\underline{v}^\alpha, \underline{P}^\alpha), \left(\overline{v}^\alpha, \overline{P}^\alpha\right)) \tag{11}$$

where $\underline{p}^\alpha = \underline{v}^\alpha p(\underline{\vartheta_i}^\alpha), \overline{p}^\alpha = \overline{v}^\alpha p(\overline{\vartheta_i}^\alpha)$,$[\vartheta_i]^\alpha = (\underline{\vartheta_i}^\alpha, \overline{\vartheta_i}^\alpha)$.

Similar with the fuzzy numbers [26-29], Z -numbers are also incorporated with three primary operations: $\oplus, \ominus$ and $\odot$. These operations are exhibited by: sum subtract multiply and division. The operations in this paper are different definitions with [20]. The α -level of Z -numbers is applied to simplify the operations.

Let us consider $Z_1 = (v_1, p_1)$ and $Z_2 = (v_2, p_2)$ be two discrete Z -numbers illustrating the uncertain variables ϑ_1 and ϑ_2, $\sum_{k=1}^{n} p_1(\vartheta_{1k}) = 1$, $\sum_{k=1}^{n} p_2(\vartheta_{2k}) = 1$. The operations are defined

$$Z_{12} = Z_1 * Z_2 = (v_1 * v_2, p_1 * p_2)$$

where $* \in \{\oplus, \ominus, \odot\}$.
The operations for the fuzzy numbers are defined as [26]

$$\begin{aligned}
[v_1 \oplus v_2]^\alpha &= [\underline{v_1}^\alpha + \underline{v_2}^\alpha, \overline{v_1}^\alpha + \overline{v_2}^\alpha] \\
[v_1 \ominus v_2]^\alpha &= [\underline{v_1}^\alpha - \underline{v_2}^\alpha, \overline{v_1}^\alpha - \overline{v_2}^\alpha] \\
[v_1 \odot v_2]^\alpha &= \begin{pmatrix} \min\{\underline{v_1}^\alpha \underline{v_2}^\alpha, \underline{v_1}^\alpha \overline{v_2}^\alpha, \overline{v_1}^\alpha \underline{v_2}^\alpha, \overline{v_1}^\alpha \overline{v_2}^\alpha\} \\ \max\{\underline{v_1}^\alpha \underline{v_2}^\alpha, \underline{v_1}^\alpha \overline{v_2}^\alpha, \overline{v_1}^\alpha \underline{v_2}^\alpha, \overline{v_1}^\alpha \overline{v_2}^\alpha\} \end{pmatrix}
\end{aligned} \tag{13}$$

For all $p_1 * p_2$ operations, we use convolutions for the discrete probability distributions

$$p_1 * p_2 = \sum_i p_1(\vartheta_{1,i}) p_2(\vartheta_{2,(n-i)}) = p_{12}(\vartheta)$$

The above definitions satisfy the Hukuhara difference [30-32],

$$\begin{aligned}
Z_1 \ominus_H Z_2 &= Z_{12} \\
Z_1 &= Z_2 \oplus Z_{12}
\end{aligned}$$

Here if $Z_1 \ominus_H Z_2$ prevails, the α -level is

$$[Z_1 \ominus_H Z_2]^\alpha = [\underline{Z}_1^\alpha - \underline{Z}_2^\alpha, \overline{Z}_1^\alpha - \overline{Z}_2^\alpha]$$

Obviously, $Z_1 \ominus_H Z_1 = 0$, $Z_1 \ominus_H Z_1 \neq 0$.
Also the above definitions satisfy the generalized Hukuhara difference [33]

$$Z_1 \ominus_{gH} Z_2 = Z_{12} \Leftrightarrow \begin{cases} 1)\ Z_1 = Z_2 \oplus Z_{12} \\ 2)\ Z_2 = Z_1 \oplus (-1) Z_{12} \end{cases} \tag{14}$$

It is convenient to display that 1) and 2) in combination are genuine if and only if Z_{12} is a crisp number. With respect to α -level what we got are $[Z_1 \ominus_{gH} Z_2]^\alpha = [\min\left\{\underline{Z}_1^\alpha - \underline{Z}_2^\alpha, \overline{Z}_1^\alpha - \overline{Z}_2^\alpha\right\}, \max\{\underline{Z}_1^\alpha - \underline{Z}_2^\alpha, \overline{Z}_1^\alpha - \overline{Z}_2^\alpha\}]$ and If $Z_1 \ominus_{gH} Z_2$ and $Z_1 \ominus_H Z_2$ subsist, $Z_1 \ominus_H Z_2 = Z_1 \ominus_{gH} Z_2$. The circumstances for the inerrancy of $Z_{12} = Z_1 \ominus_{gH} Z_2 \in E$ are

$$\begin{aligned}
&1)\begin{cases} \underline{Z}_{12}^\alpha = \underline{Z}_1^\alpha - \underline{Z}_2^\alpha \ and\ \overline{Z}_{12}^\alpha = \overline{Z}_1^\alpha - \overline{Z}_2^\alpha \\ with\ \underline{Z}_{12}^\alpha\ increasing\ \overline{Z}_{12}^\alpha\ decreasing \underline{Z}_{12}^\alpha \leq \overline{Z}_{12}^\alpha \end{cases} \\
&2)\begin{cases} \underline{Z}_{12}^\alpha = \overline{Z}_1^\alpha - \overline{Z}_2^\alpha \ and\ \overline{Z}_{12}^\alpha = \underline{Z}_1^\alpha - \underline{Z}_2^\alpha \\ with\ \underline{Z}_{12}^\alpha\ increasing\ \overline{Z}_{12}^\alpha\ decreasing, \underline{Z}_{12}^\alpha \leq \overline{Z}_{12}^\alpha \end{cases}
\end{aligned} \tag{15}$$

where $\forall \alpha \in [0,1]$
If v is a triangular function, the absolute value of Z –number $Z = (v, p)$ is

$$|Z(\vartheta)| = (|a_1| + |b_1| + |c_1|, p(|a_2| + |b_2| + |c_2|)) \tag{16}$$

If v_1 and v_2 are triangular functions, the supremum metric for Z -numbers $Z_1 =$

(v_1, p_1) and $Z_2 = (v_2, p_2)$ is given as

$$D(Z_1, Z_2) = d(v_1, v_2) + d(p_1, p_2)$$

in this case $d(\cdot,\cdot)$ is the supremum metrics considering fuzzy sets [26]. $D(Z_1, Z_2)$ is incorporated with the following possessions:

$$\begin{aligned}
&D(Z_1 + Z, Z_2 + Z) = D(Z_1, Z_2)\\
&D(Z_2, Z_1) = D(Z_1, Z_2)\\
&D(\zeta Z_1, kZ_2) = |\zeta| D(Z_1, Z_2)\\
&D(Z_1, Z_2) \le D(Z_1, Z) + D(Z, Z_2)
\end{aligned}$$

where $\in \Re$, $Z = (v, p)$ is Z -number and v is triangle function.

Definition 5. Let $\tilde{Z}$ denotes the space of Z -numbers. The $\alpha -$ level of Z -number valued function G: $[0, a] \rightarrow \tilde{Z}$ is

$$G(v, \alpha) = [\underline{G}(v, \alpha), \overline{G}(v, \alpha)]$$

where $\in \tilde{Z}$, for each $\alpha \in [0,1]$.

With the definition of Generalized Hukuhara difference, the gH-derivative of G at v_0 is expressed as

$$\frac{d}{dt} G(v_0) = \lim_{h \to 0} \frac{1}{h} [G(v_0 + h) \ominus_{gH} G(v_0)] \tag{16}$$

In (17), $G(v_0 + h)$ and $G(v_0)$ exhibits similar style with Z_1 and Z_2 respectively included in (14).

Now we utilize the fuzzy equation (3) to model the uncertain nonlinear system (1). Modeling with fuzzy equation (or fuzzy polynomial) can be regarded as fuzzy interpolation. In this paper, we utilize the fuzzy equation (1) to model the uncertain nonlinear system (1), in such a manner that the output related to the plant s_k can approach to the desired output s_k^* ,

$$\min_{W_k} \| s_k - s_k^* \| \tag{18}$$

This modeling object can be regarded as to detect $b_{i,j}$ for the following fuzzy equation

$$s_k^* = \sum_{i=1}^{n} \sum_{j=1}^{m} b_{ij} p_i(x) q_j(y) \tag{19}$$

where $x_r = [s_{k-1}^T, s_{k-2}^T, \ldots, w_k^T, w_{k-1}^T, \ldots]^T$.

3 Z-number parameter estimation with neural networks

We design a neural network to represent the fuzzy equation (3), see Fig. 1. The inputs to the neural network are x_k and y_k , the output is Z-number Z_k. The main idea is to detect appropriate weight of neural network $b_{i,j}$ in such a manner that the output of the neural network Z_k converges to the desired output s_k^*.

The input Z-numbers x_k and y_k are first applied to α -level as in (11)

$$[x_k]^\alpha = (\underline{x}_k^\alpha, \overline{x}_k^\alpha)$$
$$[y_k]^\alpha = (\underline{y}_k^\alpha, \overline{y}_k^\alpha) \tag{20}$$

Then in the first hidden units we have

$$[\Phi_i]^\alpha = (p_i(\underline{x}_k^\alpha), p_i(\overline{x}_k^\alpha) \qquad i = 1, \dots, n)$$
$$[\Phi_j]^\alpha = (q_j\left(\underline{y}_k^\alpha\right), q_j(\overline{y}_k^\alpha) \qquad j = 1, \dots, m) \tag{21}$$

and in the second hidden units we have

$$\begin{aligned}[\Phi_{i,j}]^\alpha = \{&\textstyle\sum_{i,j\in N} \underline{\Phi_i}^\alpha \underline{\Phi_j}^\alpha + \sum_{i,j\in O} \overline{\Phi_i}^\alpha \overline{\Phi_j}^\alpha + \sum_{i,j\in Q} \underline{\Phi_i}^\alpha \overline{\Phi_j}^\alpha, \\ &\textstyle\sum_{i,j\in N'} \overline{\Phi_i}^\alpha \overline{\Phi_j}^\alpha + \sum_{i,j\in O'} \underline{\Phi_i}^\alpha \underline{\Phi_j}^\alpha + \sum_{i,j\in Q'} \overline{\Phi_i}^\alpha \underline{\Phi_j}^\alpha\}\end{aligned} \tag{22}$$

where $N = \{i,j \left| \underline{\Phi_i}^\alpha \geq 0, \underline{\Phi_j}^\alpha \geq 0\right\}$,$O = \{i,j \left| \overline{\Phi_i}^\alpha < 0, \overline{\Phi_j}^\alpha < 0\right\}$, $Q = \{i,j \left| \underline{\Phi_i}^\alpha < 0, \overline{\Phi_j}^\alpha \geq 0\right\}$, $N' = i,j \left| \overline{\Phi_i}^\alpha \geq 0, \overline{\Phi_j}^\alpha \geq 0\right\}$, $O' = i,j \left| \underline{\Phi_i}^\alpha < 0, \underline{\Phi_j}^\alpha < 0\right\}$, $Q' = i,j \left| \overline{\Phi_i}^\alpha < 0, \underline{\Phi_j}^\alpha < 0\right\}$.

The neural network output is

$$\begin{aligned}[s_k]^\alpha = \{&\textstyle\sum_{i,j\in N} \underline{\Phi_{i,j}}^\alpha \underline{\mathrm{b}_{i,j}}^\alpha + \sum_{i,j\in O} \overline{\Phi_{i,j}}^\alpha \overline{\mathrm{b}_{i,j}}^\alpha + \sum_{i,j\in Q} \underline{\Phi_{i,j}}^\alpha \overline{b_{i,j}}^\alpha, \\ &\textstyle\sum_{i,j\in N'} \overline{\Phi_{i,j}}^\alpha \overline{\mathrm{b}_{i,j}}^\alpha + \sum_{i,j\in O'} \underline{\Phi_{i,j}}^\alpha \underline{\mathrm{b}_{i,j}}^\alpha + \sum_{i,j\in Q'} \overline{\Phi_{i,j}}^\alpha \underline{\mathrm{b}_{i,j}}^\alpha\}\end{aligned} \tag{23}$$

where $N = \{i,j \left| \underline{\Phi_{i,j}}^\alpha \geq 0, \underline{\mathrm{b}_{i,j}}^\alpha \geq 0\right\}$,$O = \{i,j \left| \overline{\Phi_{i,j}}^\alpha < 0, \overline{\mathrm{b}_{i,j}}^\alpha < 0\right\}$, $Q = \{i,j \left| \underline{\Phi_{i,j}}^\alpha < 0, \overline{\mathrm{b}_{i,j}}^\alpha \geq 0\right\}$, $N' = \{i,j \left| \overline{\Phi_{i,j}}^\alpha \geq 0, \overline{\mathrm{b}_{i,j}}^\alpha \geq 0\right\}$, $O' = \{i,j \left| \underline{\Phi_{i,j}}^\alpha < 0, \underline{\mathrm{b}_{i,j}}^\alpha < 0\right\}$, $Q' = \{i,j \left| \overline{\Phi_{i,j}}^\alpha < 0, \underline{\mathrm{b}_{i,j}}^\alpha < 0\right\}$.

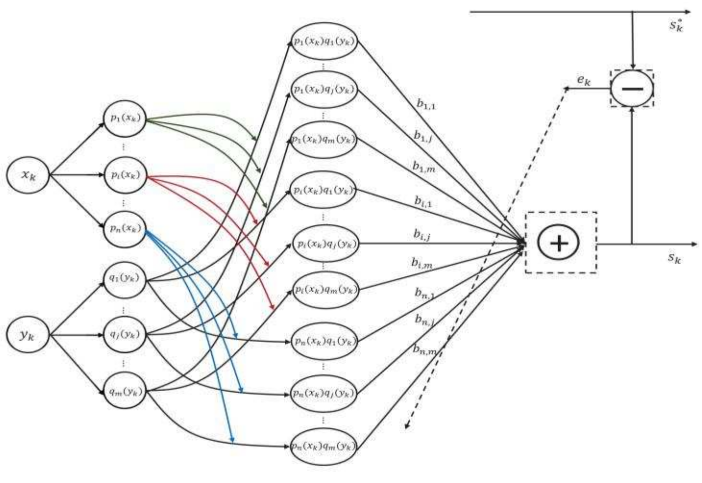

Fig. 1. Fuzzy equation in the form of neural network.

In order to train the weights, we need to define a cost function for the fuzzy numbers.

The error of the training is

$$e_k = s_k^* - s_k$$

where $[s_k^*]^\alpha = \left(\underline{s_k^*}^\alpha, \overline{s_k^*}^\alpha\right)$, $[s_k]^\alpha = \left(\underline{s_k}^\alpha, \overline{s_k}^\alpha\right)$, $[e_k]^\alpha = (\underline{e_k}^\alpha, \overline{e_k}^\alpha)$ The cost function is defined as

$$\begin{aligned} \Upsilon_k &= \underline{\Upsilon}^\alpha + \overline{\Upsilon}^\alpha \\ \underline{\Upsilon}^\alpha &= \frac{1}{2}(\underline{s_k^*}^\alpha - \overline{s_k^*}^\alpha)^2 \\ \overline{\Upsilon}^\alpha &= \frac{1}{2}(\overline{s_k^*}^\alpha - \underline{s_k^*}^\alpha)^2 \end{aligned} \tag{24}$$

Υ_k is considered to be a scalar function. It is quite obvious, $\Upsilon_k \to 0$ means $[s_k]^\alpha \to [s_k^*]^\alpha$

The vital positiveness lies within the least mean square (24) is that it has a self-correcting feature that makes it suitable to function for arbitrarily vast duration without shifting from its constraints. The mentioned gradient algorithm is subjected to cumulative series of errors and is convenient for long runs in absence of an additional error rectification procedure. It is more robust in statistics, identification and signal processing [34].

Now we use gradient method to train Z-number weight $b_{i,j} = (\underline{b_{i,j}}, \overline{b_{i,j}})$. We compute $\frac{\partial \Upsilon_k}{\partial \underline{b_{i,j}}}$ and $\frac{\partial \Upsilon_k}{\partial \overline{b_{i,j}}}$ as

$$\frac{\partial \Upsilon_k}{\partial \underline{\overline{b_{i,j}}}} = \frac{\partial \underline{\Upsilon_k}^\alpha}{\partial \underline{s_k}^\alpha}\frac{\partial \underline{s_k}^\alpha}{\partial \underline{b_{ij}}^\alpha}\frac{\partial \underline{b_{ij}}^\alpha}{\partial \underline{b_{ij}}^\alpha} + \frac{\partial \overline{\Upsilon_k}^\alpha}{\partial \overline{s_k}^\alpha}\frac{\partial \overline{s_k}^\alpha}{\partial \underline{b_{ij}}^\alpha}\frac{\partial \underline{b_{ij}}^\alpha}{\partial \underline{b_{ij}}^\alpha}$$

$$= -\left(\underline{s_k^*}^\alpha - \underline{s_k}^\alpha\right)\sum_{i,j\in N} \underline{\Phi_{i,j}}^\alpha \Gamma - \left(\overline{s_k^*}^\alpha - \overline{s_k}^\alpha\right)\left(\sum_{i,j\in O'} \underline{\Phi_{i,j}}^\alpha + \sum_{i,j\in O'} \overline{\Phi_{i,j}}^\alpha\right)\Gamma$$

where

and

$$\frac{\partial \Upsilon_k}{\partial \overline{b_{i,j}}} = \frac{\partial \underline{\Upsilon_k}^\alpha}{\partial \underline{s_k}^\alpha}\frac{\partial \underline{s_k}^\alpha}{\partial \overline{b_{ij}}^\alpha}\frac{\partial \overline{b_{ij}}^\alpha}{\partial \overline{b_{ij}}^\alpha} + \frac{\partial \overline{\Upsilon_k}^\alpha}{\partial \overline{s_k}^\alpha}\frac{\partial \overline{s_k}^\alpha}{\partial \overline{b_{ij}}^\alpha}\frac{\partial \overline{b_{ij}}^\alpha}{\partial \overline{b_{ij}}^\alpha}$$

$$= -\left(\underline{s_k^*}^\alpha - \underline{s_k}^\alpha\right)\left(\sum_{i,j\in O} \overline{\Phi_{i,j}}^\alpha + \sum_{i,j\in Q} \underline{\Phi_{i,j}}^\alpha\right)\Gamma_1 - \left(\overline{s_k^*}^\alpha - \overline{s_k}^\alpha\right)\sum_{i,j\in N'} \overline{\Phi_{i,j}}^\alpha \Gamma_1$$

where

The coefficient $b_{i,j}$ is updated as

$$\begin{aligned} \underline{b_{i,j}}(k+1) &= \underline{b_{i,j}}(k) - \eta\frac{\partial \Upsilon_k}{\partial \underline{b_{i,j}}} \\ \overline{b_{i,j}}(k+1) &= \overline{b_{i,j}}(k) - \eta\frac{\partial \Upsilon_k}{\partial \overline{b_{i,j}}} \end{aligned} \tag{25}$$

where η is the training rate $\eta > 0$. For the requirement of increasing the training process, the adding of the momentum term is mentioned as

$$\underline{b_{i,j}}(k+1) = \underline{b_{i,j}}(k) - \eta \frac{\partial \Upsilon_k}{\partial \underline{b_{i,j}}} + \Upsilon[\underline{b_{i,j}}(k) - \underline{b_{i,j}}(\mathrm{k}-1)]$$
$$\overline{b_{i,j}}(k+1) = \overline{b_{i,j}}(k) - \eta \frac{\partial \Upsilon_k}{\partial \overline{b_{i,j}}} + \Upsilon[\overline{b_{i,j}}(k) - \overline{b_{i,j}}(k-1)] \tag{25}$$

where $\gamma > 0$

Learning algorithm

1) Step 1: Choose the training rates $\eta > 0$, $\gamma > 0$ and the stop criterion $\overline{\Upsilon} > 0.$ The initial Z-number vector $B = (b_{1,1}, \dots, b_{n,m})$ is selected randomly. The initial learning iteration is $k = 1$ the initial learning error $\Upsilon = 0$.
2) Repeat the following steps for $\alpha = \alpha_1, \dots, \alpha_m$, until all training data are applied

   a) Forward calculation: Calculate the α -level of Z-number output $s_k\, S_k$ with the α -level of Z-number input vectors (x_k, y_k), and Z-number connection weight B.
   b) Back-propagation: Adjust Z-number parameters $b_{i,j}$, $i = 1, \dots, n, j = 1, \dots, m,$

      by using the cost function for the α -level of Z-number output $s_k, S_k,$ and Z-number target output s_k^* .
   c) Stop criterion: calculate the cycle error $\Upsilon_k, \Upsilon = \Upsilon + \Upsilon_k. k = k + 1$ If $\Upsilon > \overline{\Upsilon}$ let $\Upsilon = 0$, a new training cycle is initiated. Go to (a).

Conclusion

In this paper, the classical fuzzy equation is modified such that its coefficients and variables are Z-numbers. However, the parameters of the fuzzy equations cannot be obtained directly. We use the neural network method to approximate Z-number coefficients of the fuzzy equations. The neural model is constructed with the structure of fuzzy equations. With modified backpropagation method, the neural network is trained. Further work is to study the stability of training algorithms.

References

1. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids, Adv Comput Math. 12, 273-88 (2000).
2. Jafarian, A., Jafari, R., Mohamed Al Qurashi, M., Baleanud, D.: A novel computational approach to approximate fuzzy interpolation polynomials, SpringerPlus. (2016), 5:1428. doi:10.1186/s40064-016-3077-5.
3. Neidinger, R. D.: Multi variable interpolating polynomials in newton forms, in Proceedings of the JointmMathematics Meetings, 5-8, Washington, DC, USA, (2009).
4. Schroeder, H., Murthy, V.K., Krishnamurthy, E. V.: Systolic algorithm for polynomial interpolation and related problems, Parallel Computing. 493-503 (1991).
5. Zolic, A.: Numerical mathematics, Faculty of mathematics: Belgrade. 91-97, (2008).

6. Szabados, J., Vertesi, P. Interpolation of functions, World Scientific Publishing Co., Singapore, (1990).
7. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with randominputs, SIAMJournal on Scientific Computing, 27, 1118–1139, (2005).
8. Friedman, M., Ming, M., Kandel, A.: Fuzzy linear systems, Fuzzy Sets Syst. 96, 201-209 (1998).
9. Abbasbandy, S.: The application of homotopy analysis method to nonlinear equations arising in heat transfer, Phys. Lett. A, 360, 109-113 (2006).
10. Abbasbandy, S., Ezzati, R.: Newton's method for solving a system of fuzzy nonlinear equations, Appl. Math. Comput. 175, 1189-1199 (2006).
11. Allahviranloo, T. Ahmadi, N., Ahmadi, E.: Numerical solution of fuzzy differential equations by predictor-corrector method, Inform. Sci. 177, 1633-1647 (2007).
12. Kajani, M., Asady, B., Vencheh, A.: An iterative method for solving dual fuzzy nonlinear equations. Appl. Math. Comput. 167, 316-323 (2005).
13. Waziri, M., Majid, Z.: A new approach for solving dual fuzzy nonlinear equations using Broyden's and Newton's methods, Advances in Fuzzy Systems, (2012), Article 682087, 5 pages, (2012).
14. S. Pederson, M. Sambandham, The Runge-Kutta method forhybrid fuzzy differential equation, Nonlinear Anal. Hybrid Syst. 2, 626-634 (2008).
15. Buckley, J. Eslami, E.: Neural net solutions to fuzzy problems: The quadratic equation, Fuzzy Sets Syst. 86, 289-298, 1997.
16. Jafarian, A., Jafari, R., Khalili, A., Baleanud, D.: Solving fully fuzzy polynomials using feed-back neural networks, International Journal of Computer Mathematics, 92(4), 742-755 (2015).
17. Jafarian, A., Jafari, R.: Approximate solutions of dual fuzzy polynomials by feed-back neural networks, J. Soft Comput. Appl. (2012), doi:10.5899/2012/jsca-00005.
18. Mosleh, M. Evaluation of fully fuzzy matrix equations by fuzzy neural network, Appl. Math. Model. 37, 6364-6376 (2013).
19. Allahviranloo, T., Otadi, M., Mosleh, M.: Iterative method for fuzzy equations, Soft Computing, 12, 935-939 (2007).
20. Zadeh, L. A.: Toward a generalized theory of uncertainty (GTU) an outline, Inform. Sci. vol.172, pp.1-40, 2005.
21. Gardashova, L.A.: Application of operational approaches to solving decision making problem using Z-Numbers, Journal of Applied Mathematics. 5, 1323-1334 (2014).
22. Aliev, R.A., Alizadeh, A.V., Huseynov, O.H.: The arithmetic of discrete Z-numbers, Inform. Sci. 290, 134-155 (2015).
23. Kang, B., Wei, D., Li, Y., Deng, Y.: Decision making using Z--Numbers under uncertain environment, Journal of Computational Information Systems. 8, 2807-2814 (2012).
24. Kang, B., Wei, D., Li, Y. Deng, Y.: A method of converting Z-number to classical fuzzy number, Journal of Information and Computational Science. 9, 703-709 (2012).
25. Zadeh, L. A.: A note on Z-numbers, Information Sciences, 181, 2923-2932, (2011).
26. Jafari, R., Yu, W.: Fuzzy Control for Uncertainty Nonlinear Systems with Dual Fuzzy Equations, J. Intell. Fuzzy. Syst. 29, 1229-1240 (2015).
27. Jafari, R., Yu, W.: Uncertainty nonlinear systems modeling with fuzzy equations, In Proceedings of the 16th IEEE International Conference on Information Reuse and Integration, 182-188, San Francisco, Calif, USA, August, (2015).
28. Jafari, R., Yu, W.: Uncertainty Nonlinear Systems Control with Fuzzy Equations, IEEE International Conference on Systems, Man, and Cybernetics, 2885-2890, (2015).

29. Razvarz, S., Jafari, R., Granmo, O.Ch., Gegov, A.: Solution of dual fuzzy equations using a new iterative method, Asian Conference on Intelligent Information and Database Systems, 245-255, (2018).
30. Aliev, R. A., Pedryczb, W., Kreinovich, V., Huseynov, O.H.: The general theory of decisions, Inform. Sci. 327, 125-148 (2016).
31. Jafari, R., Yu, W., Li, X.: Solving Fuzzy differential equation with Bernstein neural networks, IEEE International Conference on Systems, Man, and Cybernetics, Budapest, Hungary, 1245-1250, (2016).
32. Jafari, R., Yu, W., Li, X., Razvarz, S.: Numerical solution of fuzzy differential equations with Z-numbers using Bernstein neural networks, International Journal of Computational Intelligence Systems, 10, 1226-1237, (2017).
33. Suykens, J.A.K., Brabanter, J.De., Lukas, L., Vandewalle, J.: Weighted least squares support vector machines: robustness and sparse approximation, Neurocomputing, 48,85-105 (2002).
34. Bede, B., Stefanini, L.: Generalized differentiability of fuzzy-valued functions, Fuzzy Sets Syst. 230, 119-141 (2013).