

A new computational method for solving fully fuzzy nonlinear systems

Raheleh Jafari¹, Sina Razvarz², and Alexander Gegov³

¹ Center for Artificial Intelligence Research (CAIR), University of Agder, Grimstad, Norway,

jafari3339@yahoo.com

² Departamento de Control Automático CINVESTAV-IPN (National Polytechnic Institute) Mexico City, Mexico

³ School of Computing, University of Portsmouth, Buckingham Building, Portsmouth PO13HE, UK

Abstract. Predicting the solution of complex systems is a significant challenge. Complexity is caused mainly by uncertainty and nonlinearity. The nonlinear nature of many complex systems leaves uncertainty irreducible in many cases.

In this work, a novel iterative strategy based on the feedback neural network is recommended to obtain the approximated solutions of the fully fuzzy nonlinear system (FFNS). In order to obtain the estimated solutions, a gradient descent algorithm is suggested for training the feedback neural network. An example is laid down in order to demonstrate the high accuracy of this suggested technique.

Keywords: Approximate solution, complex system, feedback neural network, gradient descent algorithm, simulation

1 Introduction

Neural network has drawn attention since it is viewed as to be successful in various applications [1–10]. Fuzzy neural network is coordinated with numerable characteristics expressed as learning capability, generalization and nonlinear mapping [11]. The standard neural network is viewed as to be approximator [12, 13]. A new learning algorithm for training the fuzzy neural networks by taking into consideration of triangular fuzzy weights is suggested in [14]. The fuzzy delta learning rule in order to train the fuzzy neural network is proposed in [15]. Obtaining the solution of fuzzy problem by utilizing neural networks is investigated in [16]. The approximated solution of fully fuzzy matrix equation by using neural networks is studied in [17]. A static neural network is suggested in [18] in order to obtain the approximate solution of fuzzy polynomials. A dynamic neural network is utilized in order to extract the estimated solution of dual fuzzy polynomials in [19]. New methods for fuzzy identification and modeling are proposed in [20–22]. The state space model of a linear system is extended to fuzzy case in [23]. The homotopy method for solving fuzzy nonlinear system is suggested in [24]. A new algorithm for dynamical nonsingleton fuzzy control system

is proposed in [25]. The Adomian methodology for solving fuzzy system of linear equations is suggested in [26]. The homotopy technique in order to obtain the solution of a system of fuzzy nonlinear equations is suggested in [27].

It is worthwhile to mention that no study has been initiated for solving FFNS by using neural networks. For this purpose, in this work a new method based on neural network is suggested in order to obtain the Z-number solutions of FFNS. The Z-number weights are adjusted by utilizing a learning algorithm which is based on the gradient descent method. The simulation results shows that the new proposed technique is effective in extracting the Z-number solutions of FFNS.

This work is organized as follows: In section II some basic definitions are laid down. The structure of the feedback neural network for obtaining the Z-number solutions of FFNS is illustrated in section III. In section IV an example with application is given in order to show the efficiency of the proposed method. The conclusion is given in Section V.

2 Preliminaries

Prior to the introduction of the FFNS, some important definitions associated with fuzzy numbers and Z-numbers are given in this section.

Definition 1. A fuzzy number B is a function $B \in E : \mathfrak{R} \rightarrow [0, 1]$, in such a manner that, 1) B is normal, (there exist $a_0 \in \mathfrak{R}$ in such a manner that $B(a_0) = 1$); 2) B is convex, $B(\varrho a + (1 - \varrho)\tau) \geq \min\{B(a), B(\tau)\}$, $\forall a, \tau \in \mathfrak{R}, \forall \varrho \in [0, 1]$; 3) B is upper semi-continuous on \mathfrak{R} , i.e., $B(a) \leq B(a_0) + \varepsilon$, $\forall a \in N(a_0)$, $\forall a_0 \in \mathfrak{R}$, $\forall \varepsilon > 0$, $N(a_0)$ is a neighborhood; 4) The set $B^+ = \{a \in \mathfrak{R}, B(a) > 0\}$ is compact.

The popular membership functions for fuzzy numbers are the triangular function

$$\mu_B = F(\lambda_1, \lambda_2, \lambda_3) = \begin{cases} \frac{a-\lambda_1}{\lambda_2-\lambda_1} & \lambda_1 \leq a \leq \lambda_2 \\ \frac{\lambda_3-a}{\lambda_3-\lambda_2} & \lambda_2 \leq a \leq \lambda_3 \end{cases} \quad (1)$$

otherwise $\mu_B = 0$, and trapezoidal function

$$\mu_B = F(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = \begin{cases} \frac{a-\lambda_1}{\lambda_2-\lambda_1} & \lambda_1 \leq a \leq \lambda_2 \\ \frac{\lambda_4-a}{\lambda_4-\lambda_3} & \lambda_3 \leq a \leq \lambda_4 \\ 1 & \lambda_2 \leq a \leq \lambda_3 \end{cases} \quad (2)$$

otherwise $\mu_B = 0$.

Definition 2. A Z-number has two components $Z = [B(a), \tilde{p}]$. The primary component $B(a)$ is defined as restriction on a real-valued uncertain variable a . The secondary component \tilde{p} is defined as a measure of reliability of B .

The probability measure is

$$\tilde{P} = \int_{\mathfrak{R}} \mu_B(a) \tilde{p}(a) da \quad (3)$$

where \tilde{p} is denoted as the probability density of a , also \mathfrak{R} is denoted as the restriction on \tilde{p} . For discrete Z-numbers

$$\tilde{P}(B) = \sum_{i=1}^n \mu_B(a_i) \tilde{p}(a_i) \quad (4)$$

Definition 3. The α -level of the fuzzy number B is illustrated as ([28]),

$$[B]^\alpha = \{a \in \mathfrak{R} : B(a) \geq \alpha\} \quad (5)$$

where $0 < \alpha \leq 1$, $B \in E$.

Since $\alpha \in [0, 1]$, $[B]^\alpha$ is bounded which is defined as $\underline{B}^\alpha \leq [B]^\alpha \leq \overline{B}^\alpha$. The α -level of B between \underline{B}^α and \overline{B}^α is stated as

$$[B]^\alpha = \left(\underline{B}^\alpha, \overline{B}^\alpha \right) \quad (6)$$

Definition 4. The Z-numbers have three elementary operations; \oplus , \ominus , and \odot which are named as addition, subtraction, and multiplication.

Suppose $Z_1 = (B_1, \tilde{p}_1)$ and $Z_2 = (B_2, \tilde{p}_2)$ be two discrete Z-numbers expressing the uncertain variables a_1 and a_2 , $\sum_{i=1}^n \tilde{p}_1(a_{1i}) = 1$, $\sum_{i=1}^n \tilde{p}_2(a_{2i}) = 1$. The operations are displayed as

$$Z_{12} = Z_1 * Z_2 = (B_1 * B_2, \tilde{p}_1 * \tilde{p}_2) \quad (7)$$

where $* \in \{\oplus, \ominus, \odot\}$.

We have

$$[B_1 \oplus B_2]^\alpha = [B_1]^\alpha + [B_2]^\alpha = [\underline{B}_1^\alpha + \underline{B}_2^\alpha, \overline{B}_1^\alpha + \overline{B}_2^\alpha] \quad (8)$$

$$[B_1 \ominus B_2]^\alpha = [B_1]^\alpha - [B_2]^\alpha = [\underline{B}_1^\alpha - \underline{B}_2^\alpha, \overline{B}_1^\alpha - \overline{B}_2^\alpha] \quad (9)$$

$$[B_1 \odot B_2]^\alpha = \left(\min\{\underline{B}_1^\alpha \underline{B}_2^\alpha, \underline{B}_1^\alpha \overline{B}_2^\alpha, \overline{B}_1^\alpha \underline{B}_2^\alpha, \overline{B}_1^\alpha \overline{B}_2^\alpha\}, \max\{\underline{B}_1^\alpha \underline{B}_2^\alpha, \underline{B}_1^\alpha \overline{B}_2^\alpha, \overline{B}_1^\alpha \underline{B}_2^\alpha, \overline{B}_1^\alpha \overline{B}_2^\alpha\} \right) \quad (10)$$

For all $\tilde{p}_1 * \tilde{p}_2$ operations, we have

$$\tilde{p}_1 * \tilde{p}_2 = \sum_i \tilde{p}_1(a_{1,i}) \tilde{p}_2(a_{2,(n-i)}) = \tilde{p}_{12}(a) \quad (11)$$

Definition 5. Absolute value of a triangular fuzzy number $B(a) = F(\lambda_1, \lambda_2, \lambda_3)$ is defined as

$$|B(a)| = |\lambda_1| + |\lambda_2| + |\lambda_3| \quad (12)$$

3 Numerical solution of fully fuzzy nonlinear system with feedback neural network

In this section a new method based on feedback neural network is proposed in order to obtain the numerical solutions of FFNS.

3.1 Fully fuzzy nonlinear systems

The FFNS is stated as below

$$\begin{cases} S_{11} \odot p \odot q \oplus \dots \oplus S_{1n} \odot p^n \odot q^n = G_1 \\ S_{21} \odot p \odot q \oplus \dots \oplus S_{2n} \odot p^n \odot q^n = G_2 \end{cases} \quad (13)$$

where $S_{1j}, S_{2j}, p, q, G_1, G_2$ belong to Z-number set (for $j = 1, \dots, n$). In order to obtain estimated solutions, a feedback neural network which is equivalent to Eq. (13) is suggested. The proposed network is shown in Figure 1.

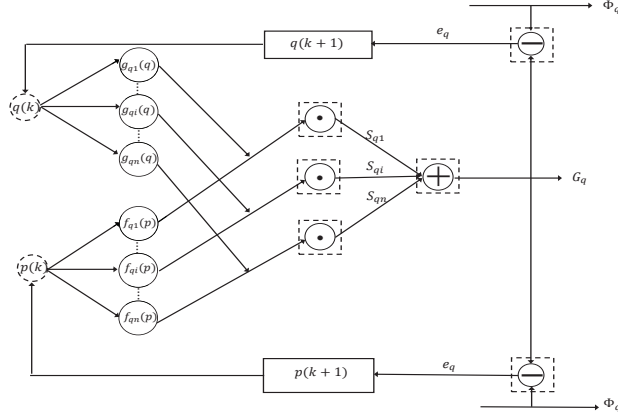


Figure 1. Fully fuzzy nonlinear system in the form of feedback neural network.

3.2 Computation of fuzzy output

We propose a four layer feedback neural network where, the α -level sets of the Z-number parameters S_{qj} are nonnegative, i.e., $0 \leq \underline{S}_{qj}^\alpha \leq \overline{S}_{qj}^\alpha$ where $j = 1, \dots, n$ and $q = 1, 2$. We have

- Input units

$$\begin{aligned} [p]^\alpha &= (\underline{p}^\alpha, \overline{p}^\alpha) \\ [q]^\alpha &= (\underline{q}^\alpha, \overline{q}^\alpha) \end{aligned} \quad (14)$$

- First hidden units

$$\begin{aligned} [u_j]^\alpha &= (\underline{u}_j^\alpha, \overline{u}_j^\alpha), \quad j = 1, \dots, n \\ [v_j]^\alpha &= (\underline{v}_j^\alpha, \overline{v}_j^\alpha), \quad j = 1, \dots, n \end{aligned} \quad (15)$$

where

$$\underline{u}_j^\alpha = \begin{cases} \underline{p}^\alpha (\underline{p}^\alpha)^{j-1}, & \underline{p}^\alpha \geq 0 \\ \overline{p}^\alpha (\overline{p}^\alpha)^{j-1}, & \underline{p}^\alpha < 0, \text{ j is even} \\ \underline{p}^\alpha (\underline{p}^\alpha)^{j-1}, & \underline{p}^\alpha < 0, \text{ j is odd} \end{cases} \quad (16)$$

$$\overline{u}_j^\alpha = \begin{cases} \overline{p}^\alpha (\overline{p}^\alpha)^{j-1}, & \overline{p}^\alpha \geq 0 \\ \underline{p}^\alpha (\underline{p}^\alpha)^{j-1}, & \overline{p}^\alpha < 0, \text{ j is even} \\ \overline{p}^\alpha (\overline{p}^\alpha)^{j-1}, & \overline{p}^\alpha < 0, \text{ j is odd} \end{cases} \quad (17)$$

$$\underline{v}_j^\alpha = \begin{cases} \underline{q}^\alpha (\underline{q}^\alpha)^{j-1}, & \underline{q}^\alpha \geq 0 \\ \overline{q}^\alpha (\overline{q}^\alpha)^{j-1}, & \underline{q}^\alpha < 0, \text{ j is even} \\ \underline{q}^\alpha (\underline{q}^\alpha)^{j-1}, & \underline{q}^\alpha < 0, \text{ j is odd} \end{cases} \quad (18)$$

$$\overline{v}_j^\alpha = \begin{cases} \overline{q}^\alpha (\overline{q}^\alpha)^{j-1}, & \overline{q}^\alpha \geq 0 \\ \underline{q}^\alpha (\underline{q}^\alpha)^{j-1}, & \overline{q}^\alpha < 0, \text{ j is even} \\ \overline{q}^\alpha (\overline{q}^\alpha)^{j-1}, & \overline{q}^\alpha < 0, \text{ j is odd} \end{cases} \quad (19)$$

• Second hidden units

$$[o_j]^\alpha = (\underline{o}_j^\alpha, \overline{o}_j^\alpha), \quad j = 1, \dots, n \quad (20)$$

where

$$\underline{o}_j^\alpha = \begin{cases} \underline{u}_j^\alpha \underline{v}_j^\alpha, & \underline{u}_j^\alpha \geq 0, \underline{v}_j^\alpha \geq 0 \\ \underline{u}_j^\alpha \overline{v}_j^\alpha, & \underline{u}_j^\alpha < 0, \underline{v}_j^\alpha \geq 0 \\ \overline{u}_j^\alpha \overline{v}_j^\alpha, & \underline{u}_j^\alpha < 0, \underline{v}_j^\alpha < 0 \\ \overline{u}_j^\alpha \underline{v}_j^\alpha, & \underline{u}_j^\alpha \geq 0, \underline{v}_j^\alpha < 0 \end{cases} \quad (21)$$

and

$$\overline{o}_j^\alpha = \begin{cases} \overline{u}_j^\alpha \overline{v}_j^\alpha, & \overline{u}_j^\alpha \geq 0, \overline{v}_j^\alpha \geq 0 \\ \overline{u}_j^\alpha \underline{v}_j^\alpha, & \overline{u}_j^\alpha < 0, \overline{v}_j^\alpha \geq 0 \\ \underline{u}_j^\alpha \overline{v}_j^\alpha, & \overline{u}_j^\alpha \geq 0, \overline{v}_j^\alpha < 0 \\ \underline{u}_j^\alpha \underline{v}_j^\alpha, & \overline{u}_j^\alpha < 0, \overline{v}_j^\alpha < 0 \end{cases} \quad (22)$$

• Output unit

$$[\Phi_q]^\alpha = (\underline{\Phi}_q^\alpha, \overline{\Phi}_q^\alpha), \quad q = 1, 2 \quad (23)$$

where

$$[\Phi_q]^\alpha = \left(\frac{\sum_{j \in M} \underline{S}_{qj}^\alpha \underline{o}_j^\alpha + \sum_{j \in N} \overline{S}_{qj}^\alpha \underline{o}_j^\alpha}{\sum_{j \in C} \overline{S}_{qj}^\alpha \overline{o}_j^\alpha + \sum_{j \in D} \underline{S}_{qj}^\alpha \overline{o}_j^\alpha} \right) \quad (24)$$

where $M = \{j | \underline{o}_j^\alpha \geq 0\}$, $N = \{j | \underline{o}_j^\alpha < 0\}$, $C = \{j | \overline{o}_j^\alpha \geq 0\}$, $D = \{j | \overline{o}_j^\alpha < 0\}$.

A cost function for α -level sets of the Z-number output Φ_q and the corresponding target output G_q is defined as below

$$\begin{aligned} e_q &= \underline{e}_q^\alpha + \overline{e}_q^\alpha \\ \underline{e}_q^\alpha &= \frac{1}{2} (G_q^\alpha - \underline{\Phi}_q^\alpha)^2 \\ \overline{e}_q^\alpha &= \frac{1}{2} (G_q^\alpha - \overline{\Phi}_q^\alpha)^2 \end{aligned} \quad (25)$$

3.3 Learning algorithm

Z-number quantities $p = ((p^1, p^2, p^3, p^4), p)$ as well as $q = ((q^1, q^2, q^3, q^4), p)$ are initialized at random Z-numbers. For Z-number variable p adjust rule is stated as below ([14, 29]),

$$\begin{aligned} p^r(k+1) &= p^r(k) \oplus \Delta p^r(k), \quad r = 1, 2, 3, 4 \\ \Delta p^r(k) &= -\eta \frac{\partial e_q}{\partial p^r} \oplus \gamma \Delta p^r(k-1) \end{aligned} \quad (26)$$

where η is the learning rate and γ is the momentum term constant. $\frac{\partial e_q}{\partial p^r}$ is computed as below

$$\frac{\partial e_q}{\partial p^r} = \frac{\partial e_q^\alpha}{\partial p^r} + \frac{\partial \bar{e}_q^\alpha}{\partial p^r} \quad (27)$$

Therefore,

$$\frac{\partial e_q^\alpha}{\partial p^r} = \frac{\partial e_q^\alpha}{\partial \Phi_q^\alpha} \frac{\partial \Phi_q^\alpha}{\partial p^r} = -(\underline{G}_q^\alpha - \underline{\Phi}_q^\alpha) \frac{\partial \Phi_q^\alpha}{\partial p^r} \quad (28)$$

where

$$\begin{aligned} \frac{\partial \Phi_q^\alpha}{\partial p^r} &= \sum_{j \in M} \frac{\partial \Phi_q^\alpha}{\partial \sigma_j^\alpha} \frac{\partial \sigma_j^\alpha}{\partial u_j^\alpha} \frac{\partial u_j^\alpha}{\partial (p^\alpha)^j} \frac{\partial (p^\alpha)^j}{\partial p^r} \\ &+ \sum_{j \in N} \frac{\partial \Phi_q^\alpha}{\partial \sigma_j^\alpha} \frac{\partial \sigma_j^\alpha}{\partial u_j^\alpha} \frac{\partial u_j^\alpha}{\partial (\bar{p}^\alpha)^j} \frac{\partial (\bar{p}^\alpha)^j}{\partial p^r} \end{aligned} \quad (29)$$

and

$$\frac{\partial \bar{e}_q^\alpha}{\partial p^r} = \frac{\partial \bar{e}_q^\alpha}{\partial \bar{\Phi}_q^\alpha} \frac{\partial \bar{\Phi}_q^\alpha}{\partial p^r} = -(\bar{G}_q^\alpha - \bar{\Phi}_q^\alpha) \frac{\partial \bar{\Phi}_q^\alpha}{\partial p^r} \quad (30)$$

where

$$\begin{aligned} \frac{\partial \bar{\Phi}_q^\alpha}{\partial p^r} &= \sum_{j \in C} \frac{\partial \bar{\Phi}_q^\alpha}{\partial \bar{\sigma}_j^\alpha} \frac{\partial \bar{\sigma}_j^\alpha}{\partial \bar{u}_j^\alpha} \frac{\partial \bar{u}_j^\alpha}{\partial (\bar{p}^\alpha)^j} \frac{\partial (\bar{p}^\alpha)^j}{\partial p^r} \\ &+ \sum_{j \in D} \frac{\partial \bar{\Phi}_q^\alpha}{\partial \bar{\sigma}_j^\alpha} \frac{\partial \bar{\sigma}_j^\alpha}{\partial \bar{u}_j^\alpha} \frac{\partial \bar{u}_j^\alpha}{\partial (\bar{p}^\alpha)^j} \frac{\partial (\bar{p}^\alpha)^j}{\partial p^r} \end{aligned} \quad (31)$$

In above relations the derivatives $\frac{\partial (p^\alpha)^j}{\partial p^r}$ and $\frac{\partial (\bar{p}^\alpha)^j}{\partial p^r}$ can be written as below

$$= \begin{cases} \frac{\partial (p^\alpha)^j}{\partial p^r} = \frac{\partial (p^\alpha)^j}{\partial p^\alpha} \frac{\partial p^\alpha}{\partial p^r} \\ j(\underline{p}^\alpha)^{j-1} \begin{cases} 1 - \alpha, & r = 1 \\ \alpha, & r = 2 \\ 0, & r = 3 \\ 0, & r = 4 \end{cases}, & \underline{p}^\alpha \geq 0 \\ j(\bar{p}^\alpha)^{j-1} \begin{cases} \alpha, & r = 2 \\ 1 - \alpha, & r = 3 \\ 0, & r = 4 \\ 0, & r = 1 \end{cases}, & \underline{p}^\alpha < 0, \quad j \text{ is even} \\ j(\underline{p}^\alpha)^{j-1} \begin{cases} \alpha, & r = 2 \\ 1 - \alpha, & r = 3 \\ 0, & r = 4 \end{cases}, & \underline{p}^\alpha < 0, \quad j \text{ is odd} \end{cases} \quad (32)$$

and

$$\begin{aligned}
& \frac{\partial(\bar{p}^\alpha)^j}{\partial p^r} = \frac{\partial(\bar{p}^\alpha)^j}{\partial \bar{p}^\alpha} \frac{\partial \bar{p}^\alpha}{\partial p^r} \\
& = \begin{cases} j(\bar{p}^\alpha)^{j-1} \begin{cases} 1 - \alpha, & r = 1 \\ \alpha, & r = 2 \\ 0, & r = 3 \\ 0, & r = 4 \end{cases}, & \bar{p}^\alpha \geq 0 \\ j(\underline{p}^\alpha)^{j-1} \begin{cases} 0, & r = 1 \\ \alpha, & r = 2 \\ 1 - \alpha, & r = 3 \\ 0, & r = 4 \end{cases}, & \bar{p}^\alpha < 0, \quad j \text{ is even} \\ j(\bar{p}^\alpha)^{j-1} \begin{cases} 0, & r = 1 \\ \alpha, & r = 2 \\ 1 - \alpha, & r = 3 \\ 0, & r = 4 \end{cases}, & \bar{p}^\alpha < 0, \quad j \text{ is odd} \end{cases} \quad (33)
\end{aligned}$$

The connection weights p_j are updated as follows

$$p_j(k+1) = (p(k+1))^j, \quad j = 2, \dots, n \quad (34)$$

We can adjust the Z-number parameter q like p .

4 Applications

In this section, a real example is used to demonstrate how to apply feedback neural network in order to find the solutions of FFNS.

Example 1 Consider the series circuit consisting a voltage source and two bulbs, see Figure 2. The power of bulbs is defined as a function of V and I . $P_1(v, i) = VI$ is power of first bulb and $P_2(v, i) = V^2I^2$ is the power of second bulb, where V is the Voltage and I is the current of bulbs having Z-number amount. The total power equation for circuit is defined as follow

$$\begin{cases} ((7, 9, 11), p(0.7, 0.81, 0.9)) \odot V \odot I \oplus ((2, 5, 7), p(0.7, 0.8, 0.9)) \odot V^2 \odot I^2 \\ = ((29, 1201, 6011), p(0.7, 0.8, 0.91)) \\ ((9, 10, 12), p(0.7, 0.8, 0.9)) \odot V \odot I \oplus ((4, 6, 9), p(0.7, 0.85, 0.9)) \odot V^2 \odot I^2 \\ = ((55, 1501, 8001), p(0.75, 0.8, 0.9)) \end{cases}$$

$V = ((-5, -3, -2), p(0.8, 0.9, 1))$ and $I = ((-4, -3, -1), p(0.8, 0.9, 1))$ are the exact solutions. The neural network shown in Figure 1 is utilized in order to estimate the solutions V and I . The maximum learning rate of neural network is $\eta = 0.001$. The neural network starts from $V(0) = ((-8, -6, -5), p(0.7, 0.8, 0.9))$ and $I(0) = ((-7, -6, -4), p(0.75, 0.8, 0.9))$. The approximation results are demonstrated in Table 1. The error between the approximate solution and the exact solution is demonstrated in Figure 3.

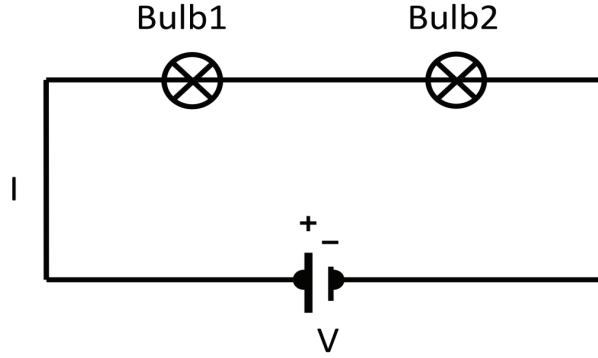


Figure 2. Series circuit

Table 1. Neural network approximation for the solutions

k	$V(k)$	$I(k)$
1	$((-7.8876, -5.9203, -4.9099), p(0.7, 0.8, 0.9))$	$((-6.8909, -5.9392, -3.9151), p(0.7, 0.8, 0.9))$
2	$((-7.5038, -5.6027, -4.5932), p(0.7, 0.8, 0.9))$	$((-6.5488, -5.6312, -3.6009), p(0.7, 0.8, 0.9))$
3	$((-7.2192, -5.3202, -4.2984), p(0.7, 0.8, 0.9))$	$((-6.2472, -5.3533, -3.2783), p(0.7, 0.8, 0.9))$
\vdots	\vdots	\vdots
97	$((-5.0107, -3.0113, -2.0127), p(0.7, 0.8, 0.9))$	$((-4.0112, -3.0106, -1.0121), p(0.7, 0.8, 0.9))$
98	$((-5.0073, -3.0086, -2.0096), p(0.7, 0.8, 0.9))$	$((-4.0089, -3.0078, -1.0093), p(0.7, 0.8, 0.9))$
99	$((-5.0043, -3.0055, -2.0062), p(0.7, 0.8, 0.9))$	$((-4.0056, -3.0049, -1.0061), p(0.7, 0.8, 0.9))$

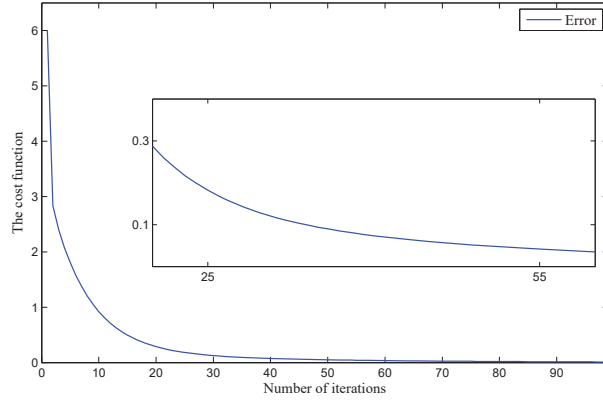


Figure 3. The error between the approximate solution and the exact solution

5 Conclusion

In this work, a new method based on the feedback neural network is proposed in order to approximate the solutions of FFNS. In order to obtain the approximate

solutions of FFNS a learning algorithm based on the gradient descent method is applied. An example is laid down in order to demonstrate the high accuracy of this suggested technique. The future work is the application of the mentioned methodology for FFNS equations on the basis of Z-numbers.

References

1. R. Jafari, W. Yu, Fuzzy Control for Uncertainty Nonlinear Systems with Dual Fuzzy Equations, *J. Intell. Fuzzy. Syst.* Vol.29, pp.1229-1240, 2015.
2. R. Jafari, W. Yu, X. Li, Fuzzy Differential Equation for Nonlinear System Modeling with Bernstein Neural Networks, *IEEE Access.* doi:10.1109/ACCESS.2017.2647920, 2017.
3. R. Jafari, W. Yu, Uncertainty Nonlinear Systems Modeling with Fuzzy Equations, *Mathematical problems in Engineering.* Vol.2017, <https://doi.org/10.1155/2017/8594738>, 2017.
4. R. Jafari, W. Yu, Uncertainty nonlinear systems modeling with fuzzy equations, *In Proceedings of the 16th IEEE International Conference on Information Reuse and Integration*, pp.182-188, San Francisco, Calif, USA, August 2015.
5. R. Jafari, W. Yu, X. Li, Numerical solution of fuzzy equations with Z-numbers using neural networks, *Intelligent automation and Soft Computing*, pp.1-7, 2017.
6. R. Jafari, W. Yu, X. Li, S. Razvarz, Numerical Solution of Fuzzy Differential Equations with Z-numbers Using Bernstein Neural Networks, *International Journal of Computational Intelligence Systems*, Vol.10, pp.1226-1237, 2017.
7. S. Razvarz, R. Jafari, W. Yu, Numerical solution of fuzzy differential equations with Z-numbers using fuzzy Sumudu Transforms, *Advances in Science, Technology and Engineering Systems Journal*, Vol.3, pp.66-75, 2018.
8. S. Razvarz, R. Jafari, O.Ch. Granmo, A. Gegov, Solution of dual fuzzy equations using a new iterative method, *Asian Conference on Intelligent Information and Database Systems*, pp.245-255, 2018.
9. M. Pratama, J. Lu, S.G. Anavatti, E. Lughofer, C.P. Lim, An Incremental Meta-cognitive-based Scaffolding Fuzzy Neural Network, *Neurocomputing*, Vol.171, pp.89-105, 2016.
10. M. Pratama, J. Lu, E. Lughofer, G. Zhang, M.J. Er, Incremental Learning of Concept Drift Using Evolving Type-2 Recurrent Fuzzy Neural Network, *IEEE Transactions on Fuzzy Systems*, Vol.25, pp.1175-1192, 2017.
11. R.H. Abiyev, O. Kaynak, Fuzzy wavelet neural networks for identification and control of dynamic plants: a novel structure and a comparative study, *IEEE Trans. Ind. Electron.* Vol.55, pp.3133-3140, 2008.
12. K. Hornik, Approximation capabilities of multilayer feed-forward networks, *Neural Networks.* Vol.4, pp.251-257, 1991.
13. F. Scarselli, A.C. Tsoi, Universal approximation using feedforward neural networks: a survey of some existing methods, and some new results, *Neural Networks.* Vol.11, pp.15-37, 1998.
14. H. Ishibuchi, K. Kwon, H. Tanaka, A learning of fuzzy neural networks with triangular fuzzy weights, *Fuzzy Sets Syst.* Vol.71, pp.277-293, 1995.
15. Y. Hayashi, J.J. Buckley, E. Czogala, Fuzzy neural network with fuzzy signals and weights, *Int. J. Intell. Syst.* Vol.8, pp.527-537, 1993.
16. J.J. Buckley, E. Eslami, Neural net solutions to fuzzy problems: The quadratic equation, *Fuzzy Sets Syst.* Vol.86, pp.289-298, 1997.

17. M. Mosleh, Evaluation of fully fuzzy matrix equations by fuzzy neural network, *Appl. Math. Model.* Vol.37, pp.6364-6376, 2013.
18. S. Abbasbandy, M. Otadi, Numerical solution of fuzzy polynomials by fuzzy neural network, *Appl. Math. Comput.* Vol.181, pp.1084-1089, 2006.
19. A. Jafarian, R. Jafari, Approximate solutions of dual fuzzy polynomials by feed-back neural networks, *Journal of Soft Computation and Applications*. doi:10.5899/2012/jsca-00005.
20. T. Takagi, M. Sugeno, identification of systems and its applications to modeling and control, *IEEE Trans. On Man and Cybernetics*. Vol.15, pp.116-132, 1985.
21. E. Lughofer, C. Cernuda, S. Kindermann, M. Pratama, Generalized Smart Evolving Fuzzy Systems, *Evolving Systems*, Vol.6, pp.269-292, 2015.
22. P.P. Angelov, D. Filev, An approach to online identification of Takagi-Sugeno fuzzy models, *IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics*, Vol.34, pp.484-498, 2004.
23. G. Chen, T.T. Pham, J.J. Weiss, modeling of control systems, *IEEE Aerosp. Electron. Syst.* Vol.31, pp.414-428, 1995.
24. S. Abbasbandy, R. Ezzati, Homotopy method for solving fuzzy nonlinear equations, *Applied Sciences*. Vol.8, pp.1-7, 2006.
25. G.C. Mouzouris, J.M. Mendel, Dynamic non-singleton logic systems for nonlinear modeling, *IEEE Trans. On Systems*. Vol.5, pp.199-208, 1997.
26. T. Allahviranloo, The Adomian decomposition method for fuzzy system of linear equations, *Applied Mathematics and Computation*. Vol.163, pp.553-563, 2005.
27. M. Paripour, E. Zarei, A. Shahsavaran, Numerical solution for a system of fuzzy nonlinear equations, *Journal of Fuzzy Set Valued Analysis*, Vo.2014, pp.1-10, 2014.
28. L.A. Zadeh, Toward a generalized theory of uncertainty (GTU) an outline, *Inform. Sci.* Vol.172, pp.1-40, 2005.
29. G. Alefeld, J. Herzberger, Introduction to Interval Computations, *Academic Press. New York*. 1983.