

RESEARCH

Open Access



# A novel computational approach to approximate fuzzy interpolation polynomials

Ahmad Jafarian<sup>1\*</sup>, Raheleh Jafari<sup>2</sup>, Maysaa Mohamed Al Qurashi<sup>3</sup> and Dumitru Baleanu<sup>4,5</sup>

\*Correspondence:  
jafarian5594@yahoo.com  
<sup>1</sup> Department  
of Mathematics, Urmia  
Branch, Islamic Azad  
University, Urmia, Iran  
Full list of author information  
is available at the end of the  
article

## Abstract

This paper build a structure of fuzzy neural network, which is well sufficient to gain a fuzzy interpolation polynomial of the form  $y_p = a_n x_p^n + \dots + a_1 x_p + a_0$  where  $a_j$  is crisp number (for  $j = 0, \dots, n$ ), which interpolates the fuzzy data  $(x_j, y_j)$  (for  $j = 0, \dots, n$ ). Thus, a gradient descent algorithm is constructed to train the neural network in such a way that the unknown coefficients of fuzzy polynomial are estimated by the neural network. The numeral experimentations portray that the present interpolation methodology is reliable and efficient.

**Keywords:** Fuzzy neural networks, Fuzzy interpolation polynomial, Cost function, Learning algorithm

## Background

Artificial neural networks (ANNs) are mathematical or computational models based on biological neural networks. Neural networks consist of universal approximation potentiality, and they function best when the system has a high endurance to error when used to model. Recently, there have been rapid growth of ANNs which was utilized in various fields (Abbasbandy and Otadi 2006; Chen and Zhang 2009; Guo and Qin 2009; Jafarian and Jafari 2012; Jafarian et al. 2015a, b; Jafarian and Measoomynia 2011, 2012; Song et al. 2013; Wai and Lin 2013). One of the vital roles of ANN is finding FIPs as it proposed in this research.

Interpolation theory is one of the basic tool in applied and numerical mathematics. Interpolation has been used extensively, because it is one of the noteworthy techniques of function approximation (Boffi and Gastaldi 2006; Mastlyo 2010; Rajan and Chaudhuri 2001). Using Newton's divided difference scheme, a new technique was established in Schroeder et al. (1991) for polynomial interpolation. The problem related to multivariate interpolation has grabbed the attention of researchers world wide (Neidinger 2009; Olver 2006). There are various multivariate interpolation methods. In Olver (2006) they used a multivariate Vandermode matrix and its LU factorization, and Neidinger (2009) utilized the Newton-form interpolation. We recall that sparse grid interpolation is a further technique. In recent years this procedure is widely executed for the provision of an average approximation to a smooth function (Xiu and Hesthaven 2005). Utilizing the Lagrange interpolating polynomials, this approach

introduces a polynomial interpolant on the basis of amounts of the function at the points in an amalgamation of product grids of minute dimension (Barthelmann et al. 2000). Existing trends on interpolation networks, have been revealed in Llanas and Sainz (2006), Sontag (1992). Numerable proof based on the notation that single hidden layer FNNs taking into account  $m + 1$  neurons, is able to learn  $m + 1$  isolated data  $(x_i, f_i)$  (for  $i = 0, \dots, m$ ) with zero error has been established in Ito (2001). The detailed introduction and survey of major results can be extracted from Refs. Szabados and Vertesi (1990), Tikhomirov (1990).

This paper is inclined to the motive in order to deliver a fuzzy modeling technique by the utilization of FNNs for finding a FIP of the form

$$y_p = a_n x_p^n + \dots + a_1 x_p + a_0, \tag{1}$$

where  $a_j \in \mathbb{R}$  (for  $j = 0, \dots, n$ ), which interpolates the fuzzy data  $(x_j, y_j)$  (for  $j = 0, \dots, n$ ). The proposed network is a formation, abiding of three layers whereas the extension principle of Zadeh (2005) elaborately describes the input-output connection of each unit. In the latest model, the unrevealed coefficients of fuzzy polynomial can be approximated by employing a cost function. Moreover, a learning technique which is associated with gradient decent procedure is formulated for the adjustment of connection weights to any achievable degree of precision.

This paper starts with a summary explanation of fuzzy numbers and fuzzy interpolation, then we provide the method of FNN for finding the crisp solution of the FIP. Two numerical examples are proposed to establish the validity and performance of the justified approach in “Numerical examples” section. Finally, “Concluding remarks” section presents the conclusions.

### Method description

Basically, the interpolation theory has a wide range of applications in mathematical analysis. In numerical analysis, the interpolation is a method or operation of finding from a few given terms of a series, as of numbers or observations, other intermediate terms in conformity with the law of the series. Generally, the interpolation techniques are in phase with elementary model of an interpolating function which can be stated as:

$$s : \mathbb{R} \rightarrow \mathbb{R},$$

$$s(x) = \sum_{j=1}^n y_j \cdot \phi_j(x), \tag{2}$$

with the basis function  $\phi_j(x) : \mathbb{R} \rightarrow \mathbb{R}$  that elates in interpolation criteria:

$$\phi_j(x_k) = \begin{cases} 1, & \text{for } k = j, \\ 0, & \text{for } k \neq j. \end{cases}$$

Suppose that  $\hat{x}_1, \dots, \hat{x}_n$  be  $n$  fuzzy points in  $E^n$  whereas, a fuzzy number  $\hat{y}_j \in E$  is in direct relation with each  $\hat{x}_j$  for  $j = 1, \dots, n$ . The sought function can be portrayed as follows:

$$\hat{s} : E^n \rightarrow E : \hat{s}(\hat{x}) = \sum_{j=1}^n \hat{y}_j \cdot \hat{\phi}_j(\hat{x}), \tag{3}$$

where the  $\hat{\phi}_j : E^n \rightarrow E$  for  $j = 1, \dots, n$  exhibit fuzzy functions that compensate the stipulation of interpolation:

$$\hat{\phi}_j(\hat{x}_k) = \begin{cases} 1, & \text{for } k = j, \\ 0, & \text{for } k \neq j. \end{cases}$$

**Fuzzy interpolation polynomial**

The interested are vested in finding FIP of the form

$$y_p = a_n x_p^n + \dots + a_1 x_p + a_0, \tag{4}$$

where  $a_j \in \mathbb{R}$  (for  $j = 0, \dots, n$ ), that interpolates the fuzzy data  $(x_j, y_j)$  (for  $j = 0, \dots, n$ ). Taking into account a three layer FNN construction which is displayed by Fig. 1. The input-output connection of each unit of the offered neural network can be portrayed as mentioned below, when the  $\alpha$ -level sets of the fuzzy input  $x_p$  is nonnegative, i.e.,  $0 \leq [x_p]_l^\alpha \leq [x_p]_u^\alpha$ :

- *Input unit*

$$[o]^\alpha = [x_p]_l^\alpha, [x_p]_u^\alpha, \quad p = 0, \dots, n. \tag{5}$$

- *Hidden units*

$$[O_j]^\alpha = f([net_j]_l^\alpha, [net_j]_u^\alpha) = \left( ([o]_l^\alpha)^j, ([o]_u^\alpha)^j \right), \quad j = 1, \dots, n. \tag{6}$$

- *Output unit*

$$\begin{aligned} [y_p]^\alpha &= F([Net]_l^\alpha + a_0, [Net]_u^\alpha + a_0) \\ &= ([Net]_l^\alpha + a_0, [Net]_u^\alpha + a_0), \quad p = 0, \dots, n, \end{aligned} \tag{7}$$

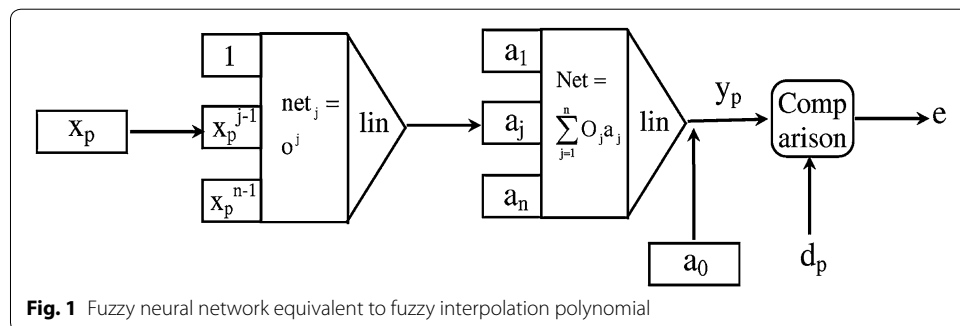
We have

$$[Net]_l^\alpha = \sum_{j \in M} [O_j]_l^\alpha \cdot a_j + \sum_{j \in C} [O_j]_u^\alpha \cdot a_j,$$

and

$$[Net]_u^\alpha = \sum_{j \in M} [O_j]_u^\alpha \cdot a_j + \sum_{j \in C} [O_j]_l^\alpha \cdot a_j,$$

where  $M = \{a_j \geq 0\}$ ,  $C = \{a_j < 0\}$  and  $M \cup C = \{1, \dots, n\}$ .



**Fig. 1** Fuzzy neural network equivalent to fuzzy interpolation polynomial

**Cost function**

The input signals  $x_p$  (for  $p = 0, \dots, n$ ) are represented to the network and then  $y_n(x_p)$  which is an representing the network output upon the presentation of  $a_j$  (for  $j = 0, \dots, n$ ), is calculated. Defining of cost function over the model parameters makes it a good forecaster. The mean squared error is termed to be as one of the vastly popular usable cost function. Now, let the  $\alpha$ -level sets of the fuzzy target output  $d_p$  are exhibited as:

$$[d_p]^\alpha = [d_p]_l^\alpha, [d_p]_u^\alpha, \quad \alpha \in [0, 1],$$

A cost function which is required to be diminished is stated for each  $\alpha$ -level sets as depicted:

$$e_p(\alpha) = e_p^l(\alpha) + e_p^u(\alpha), \quad p = 0, \dots, n, \tag{8}$$

where

$$e_p^l(\alpha) = \alpha \cdot \frac{([d_p]_l^\alpha - [y_p]_l^\alpha)^2}{2}, \tag{9}$$

$$e_p^u(\alpha) = \alpha \cdot \frac{([d_p]_u^\alpha - [y_p]_u^\alpha)^2}{2}. \tag{10}$$

Generally the summed up error of the suggested neural network is extracted by:

$$e = \sum_{\alpha} \sum_{p=0}^n e_p(\alpha). \tag{11}$$

Obviously,  $e \rightarrow 0$  means  $[y_p]^\alpha \rightarrow [d_p]^\alpha$ .

**Fuzzy neural network learning approach**

Suppose connection weights  $a_j$  (for  $j = 0, \dots, n$ ) are randomly actuated by crisp numbers. Now tweaked rule is illustrated as (Ishibuchi et al. 1995):

$$a_j(t + 1) = a_j(t) + \Delta a_j(t),$$

$$\Delta a_j(t) = -\eta \cdot \frac{\partial e_p(\alpha)}{\partial a_j} + \gamma \cdot \Delta a_j(t - 1), \tag{12}$$

where  $t$  denotes the number of moderation,  $\eta$  signifies the rate of learning and  $\gamma$  implies as the stationary momentum term. We calculate  $\frac{\partial e_p(\alpha)}{\partial a_j}$  as follows:

$$\frac{\partial e_p(\alpha)}{\partial a_j} = \frac{\partial e_p^l(\alpha)}{\partial a_j} + \frac{\partial e_p^u(\alpha)}{\partial a_j}. \tag{13}$$

Hence complexities lies in the calculation of the derivatives  $\frac{\partial e_p^l(\alpha)}{\partial a_j}$  and  $\frac{\partial e_p^u(\alpha)}{\partial a_j}$ . So we have:

$$\begin{aligned} \frac{\partial e_p^l(\alpha)}{\partial a_j} &= \frac{\partial e_p^l(\alpha)}{\partial [y_p]_l^\alpha} \cdot \frac{\partial [y_p]_l^\alpha}{\partial [Net]_l^\alpha} \cdot \frac{\partial [Net]_l^\alpha}{\partial a_j} \\ &= -\alpha \cdot ([d_p]_l^l - [y_p]_l^l) \cdot \frac{\partial [Net]_l^\alpha}{\partial a_j}, \quad j = 1, \dots, n, \end{aligned}$$

and

$$\frac{\partial e_p^l(\alpha)}{\partial a_j} = \frac{\partial e_p^l(\alpha)}{\partial [y_p]_l^\alpha} \cdot \frac{\partial [y_p]_l^\alpha}{\partial a_j} = -\alpha \cdot ([d_p]_l^l - [y_p]_l^l), \quad j = 0,$$

where

$$\frac{\partial [Net]_l^\alpha}{\partial a_j} = \begin{cases} [O_j]_l^\alpha, & a_j \geq 0, \\ [O_j]_u^\alpha, & a_j < 0, \end{cases}$$

also we have

$$\begin{aligned} \frac{\partial e_p^u(\alpha)}{\partial a_j} &= \frac{\partial e_p^u(\alpha)}{\partial [y_p]_u^\alpha} \cdot \frac{\partial [y_p]_u^\alpha}{\partial [Net]_u^\alpha} \cdot \frac{\partial [Net]_u^\alpha}{\partial a_j} \\ &= -\alpha \cdot ([d_p]_u^u - [y_p]_u^u) \cdot \frac{\partial [Net]_u^\alpha}{\partial a_j}, \quad j = 1, \dots, n, \end{aligned}$$

and

$$\frac{\partial e_p^u(\alpha)}{\partial a_j} = \frac{\partial e_p^u(\alpha)}{\partial [y_p]_u^\alpha} \cdot \frac{\partial [y_p]_u^\alpha}{\partial a_j} = -\alpha \cdot ([d_p]_u^u - [y_p]_u^u), \quad j = 0,$$

where

$$\frac{\partial [Net]_u^\alpha}{\partial a_j} = \begin{cases} [O_j]_u^\alpha, & a_j \geq 0, \\ [O_j]_l^\alpha, & a_j < 0, \end{cases}$$

### Upper bound approximation

**Theorem 1** Suppose  $p : \mathfrak{R} \rightarrow \mathfrak{R}$  is a continuous function, hence for each compact set  $\vartheta \subset E_0$  (the set of all the bounded fuzzy set), and  $\psi > 0$ , there are  $m \in \mathbb{N}$ , and  $a_0, a_i \in \mathfrak{R}, i = 1, 2, \dots, m$ , which imply

$$\forall \hat{x} \in \vartheta \quad \text{and} \quad \forall \check{x} \in \mathfrak{R}, \quad d\left(p(\check{x}), \sum_{i=1}^m p_i(\hat{x})a_i + a_0\right) < \psi, \tag{14}$$

where  $\psi$  is a finite number.

*Proof* The proof of theorem can be followed from the below results. □

If  $p : \mathfrak{X} \rightarrow \mathfrak{X}$ , by applying the methodology of the extension principle,  $p$  can be extended to the fuzzy function that denotes by  $p : E_0 \rightarrow E$  as follows:

$$\forall u \in E_0, \quad p(u)(y) = \bigvee_{p(\hat{x})=y} \{u(\hat{x})\} \quad y \in \mathfrak{X}, \tag{15}$$

$p$  is termed as expanded function. Also,  $cc(\mathfrak{X})$  implies the bounded set of closed intervals of  $\mathfrak{X}$ . clearly

$$u \in E_0 \implies \forall \alpha \in (0, 1], \quad [u]^\alpha \in cc(\mathfrak{X}). \tag{16}$$

Moreover

$$Supp(u) \in cc(\mathfrak{X}). \tag{17}$$

Henceforth, we let

$$Supp(u) = [s_1(u), s_2(u)]. \tag{18}$$

**Theorem 2** Suppose  $p : \mathfrak{X} \rightarrow \mathfrak{X}$  is a continuous function, hence for each compact set  $\vartheta \subset E_0, \varrho > 0$  and arbitrary  $\varepsilon > 0$ , exist  $m \in N$ , and  $a_0, a_i \in \mathfrak{X}, i = 1, 2, \dots, m$ , implicate

$$\forall \hat{x} \in \vartheta, \quad d\left(p(\hat{x}), \sum_{i=1}^m p_i(\hat{x})a_i + a_0\right) < \varrho, \tag{19}$$

where  $\varrho$  is a finite number. The bottom and top bounds of the  $\alpha$ -level set of fuzzy function diminish to  $\varrho$ , but the center goes to  $\varepsilon$ .

*Proof* Because  $\vartheta \subset E_0$  is a compact set, and so by Lemma 3, it can be supposed that  $V \subset \mathfrak{X}$  be the compact set associated to  $\vartheta. \forall \varepsilon > 0$ , therefore by the final outcome in Cybenko (1989), exist  $m \in N$ , and  $a_0, a_i \in \mathfrak{X}, i = 1, 2, \dots, m$ , which imply

$$\forall \hat{x} \in V, \quad \left| p(\hat{x}) - \sum_{i=1}^m p_i(\hat{x})a_i + a_0 \right| < \varepsilon, \tag{20}$$

holds. Let  $q(\hat{x}) = \sum_{i=1}^m p_i(\hat{x})a_i + a_0, \hat{x} \in \mathfrak{X}$ , then

$$\forall \hat{x} \in V, \quad |p(\hat{x}) - q(\hat{x})| < \varepsilon. \tag{21}$$

Theorem 4 implies the validity of (19). □

**Lemma 3** If  $\vartheta \subset E_0$  be a compact set, hence  $\vartheta$  is uniformly support-bounded, i.e. exists a compact set  $V \subset \mathfrak{X}$ , implicates  $\forall u \in \vartheta, Supp(u) \subset V$ .

**Theorem 4** Supposing  $\vartheta \subset E_0$  be compact,  $V$  the corresponding compact set of  $\vartheta$ , and  $p, q : \mathfrak{X} \rightarrow \mathfrak{X}$  are the continuous functions that compensate the relation mentioned below

$$\forall \hat{x} \in V, \quad |p(\hat{x}) - q(\hat{x})| < k, \quad k > 0. \tag{22}$$

Then  $\forall u \in \vartheta, d(p(u) - q(u)) \leq k$ .

*Proof* See Liu (2000). □

### Numerical examples

The following examples has been used to narrate the methodology proposed in this paper.

*Example 5* The connection between three tanks and pipeline which is denoted by a constant  $H$  is represented by Fig. 2. It is a requirement to pump water in order to transfer it from one tank to the further two tanks. The mentioned system suffice the relation mentioned below

$$H = A_0 \oplus A_1 F_1 \oplus A_2 F_2 \oplus A_3 F_3$$

here  $F_1 = \sqrt{2x}, F_2 = x\sqrt{x}, F_3 = x^3$  are considered to be the flow quantity, where  $x$  is taken to be the elapsed time. The height of the pipe is mentioned by the term  $H, A_0, A_1, A_2$  and  $A_3$  are the pump characteristic coefficients, to be mentioned

$$A_0 = 2, \quad A_1 = 4, \quad A_2 = 3, \quad A_3 = 5$$

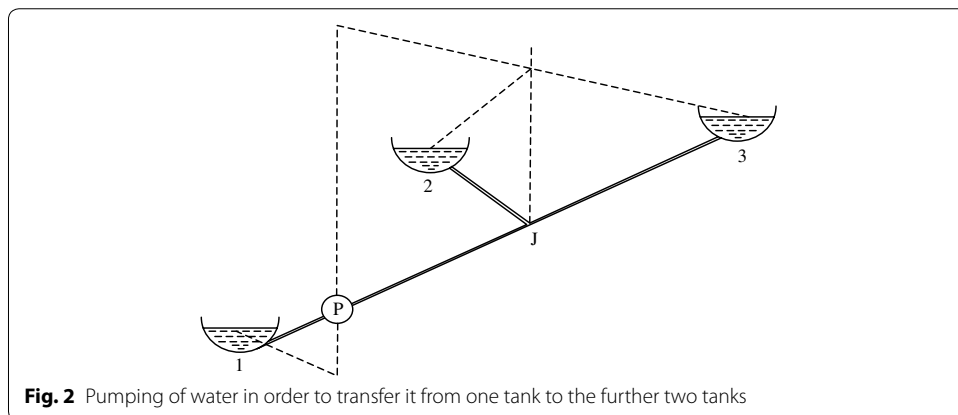
In below, four real uncertain data have been mentioned

$$x = \{6, (1, 3, 4), 3, (2, 3, 4, 6)\}$$

The iteration of data is continued for 19 times.

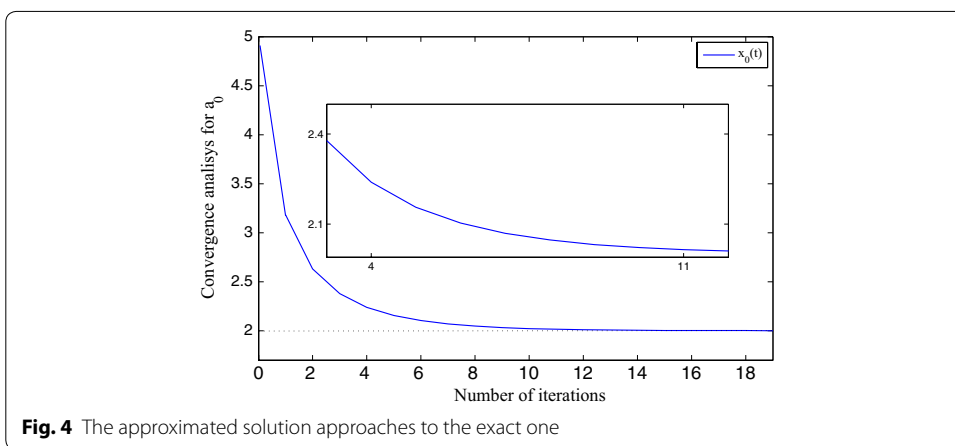
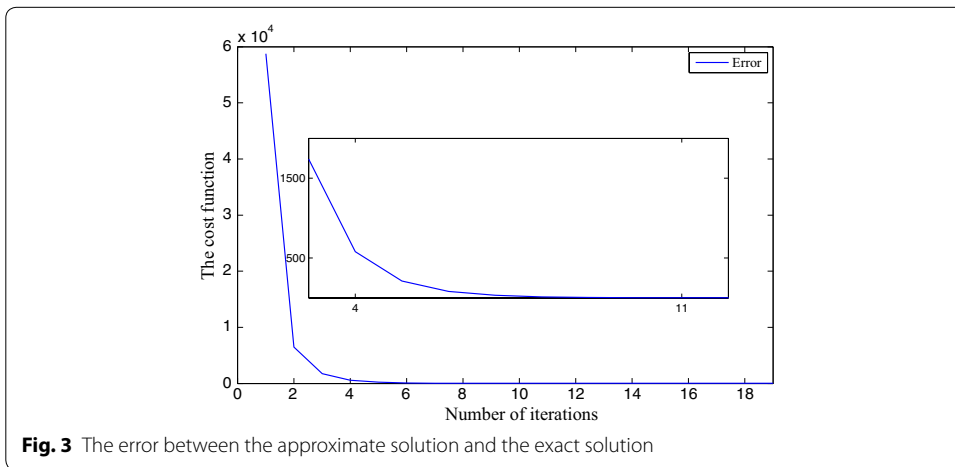
$$H = \{1139.9472, (15.6568, 162.3859, 357.3137), 162.3893, (58.4852, 162.3859, 357.3137, 1139.9456)\}$$

We use  $x_0 = 5, x_1 = 7, x_2 = 6, x_3 = 8, \eta = 1 \times 10^{-2}$  and  $\gamma = 1 \times 10^{-2}$  for FNN. The approximation results are depicted in Table 1. The precision level of the solutions  $x_0(t), x_1(t), x_2(t)$  and  $x_3(t)$  are shown in Fig. 3,  $t$  implies the iterative numbers. It is eminent that by incrementing the iterations, the cost function diminishes to zero. The convergency criteria of the approximated solutions are portrayed using Figs. 4, 5, 6 and 7. For the purpose of attaining the exact solutions, the iterations in the figures have to be increased.

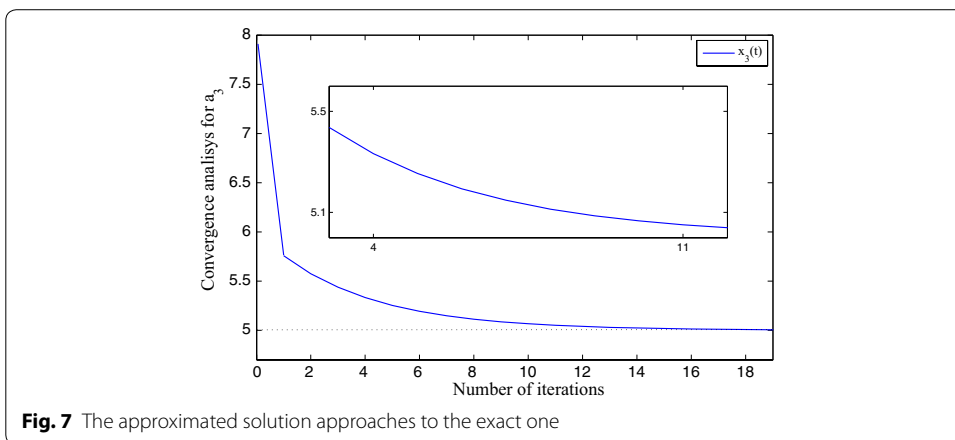
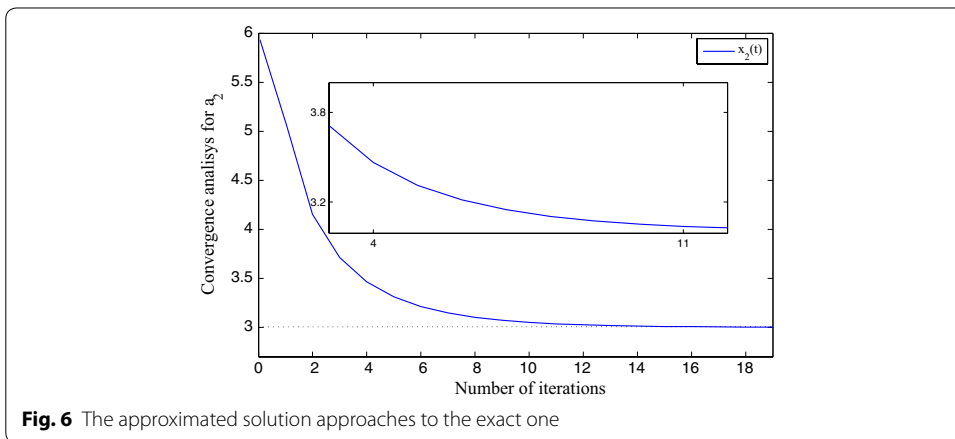
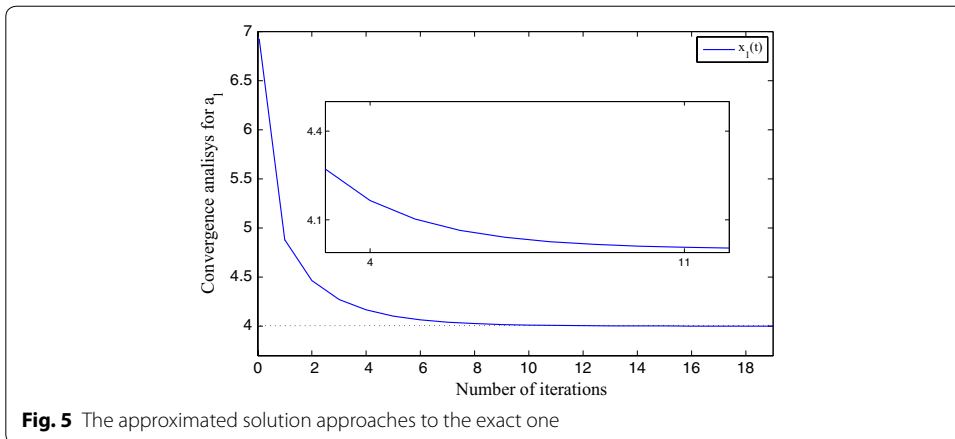


**Table 1 Neural network approximation for the coefficients**

$t$	$x_0(t)$	$x_1(t)$	$x_2(t)$	$x_3(t)$	Error for FNN
1	4.9018	6.9215	5.9307	7.9121	58,756.65
2	4.5321	6.6450	5.5480	7.6010	6479.790
3	4.0231	6.2056	5.1250	7.2212	1741.483
4	3.6850	5.8401	4.7851	6.7945	577.7597
5	3.2032	5.4001	4.3365	6.3330	210.8822
⋮	⋮	⋮	⋮	⋮	⋮
15	2.0008	4.0007	3.0008	5.0006	0.366883
16	2.0007	4.0005	3.0006	5.0005	0.151818
17	2.0005	4.0004	3.0005	5.0003	0.062895
18	2.0004	4.0003	3.0004	5.0002	0.026075
19	2.0003	4.0002	3.0003	5.0001	0.010815







*Example 6* Contemplate the sequential interpolation points:

$$((1, 2, 3); (-54, -29, -12)), ((3, 4, 6); (-177, -87, -54)),$$

$$((2, 3, 5); (-128, -54, -29))$$

The exact solution for the given problem can be stated as:

$$y = -4x^2 - 5x - 3.$$

This constrained is resolved by utilizing the technique of neural network suggested in this context, assuming  $x_0 = -0.5, x_1 = -2.5, x_2 = -1.5, \eta = 3 \times 10^{-2}$  and  $\gamma = 3 \times 10^{-2}$ .

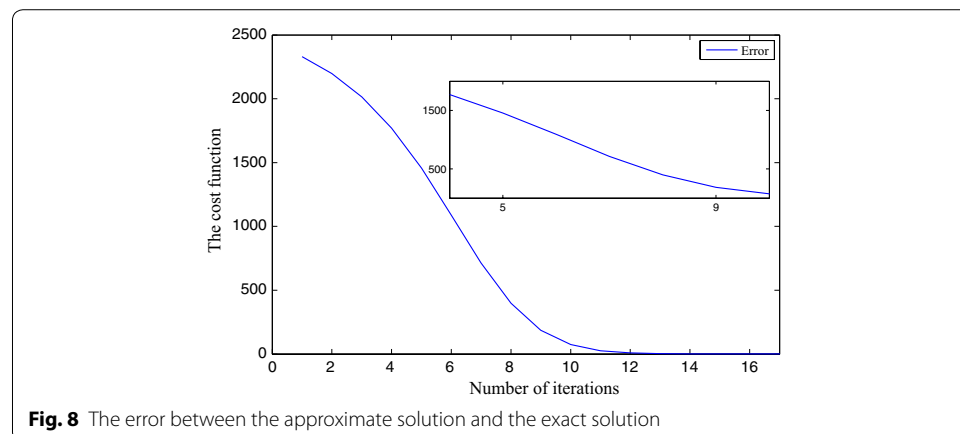
The approximation results are depicted in Table 2. The precision level of the solutions  $x_0(t), x_1(t)$  and  $x_2(t)$  are shown in Fig. 8,  $t$  implies the number of iterations.

### Concluding remarks

This research introduces a new methodology in order to find a FIP which interpolates the fuzzy data  $(x_j, y_j)$  (for  $j = 0, \dots, n$ ). To achieve this goal, a FNN equivalent to FIP was built, and a fast learning algorithm was defined for approximating the crisp unknown coefficients of the given polynomial. The proposed method was based on approximating FNN and the MATLAB software is used for the simulations. The innovative method was validated with two examples. The simulation results clearly illustrated the efficiency and computational advantages of the proposed technique. In particular, the error of approximation is minute.

**Table 2 Neural network approximation for the coefficients**

$t$	$x_0(t)$	$x_1(t)$	$x_2(t)$	Error for FNN
1	-0.5915	-2.5895	-1.5784	2330.5296
2	-0.9910	-2.9033	-1.9664	1896.6752
3	-1.3356	-3.3346	-2.3696	999.56201
4	-1.8050	-3.8798	-2.7561	401.56201
5	-2.2257	-4.1035	-3.1100	95.188500
⋮	⋮	⋮	⋮	⋮
13	-2.9996	-4.9995	-3.9996	0.86688366
14	-2.9998	-4.9996	-3.9998	0.54635274
15	-2.9999	-4.9998	-3.9999	0.23614301
16	-3.0000	-4.9999	-4.0000	0.06896850
17	-3.0000	-5.0000	-4.0000	0.02003805



**Fig. 8** The error between the approximate solution and the exact solution

**Authors' contributions**

All authors contributed equally to this work. All authors read and approve the final manuscript.

**Author details**

<sup>1</sup> Department of Mathematics, Urmia Branch, Islamic Azad University, Urmia, Iran. <sup>2</sup> Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), Mexico City, Mexico. <sup>3</sup> Department of Mathematics, King Saud University, Riyadh 11495, Saudi Arabia. <sup>4</sup> Department of Mathematics, Faculty of Art and Sciences, Cankaya University, 06530 Balgat, Ankara, Turkey. <sup>5</sup> Institute of Space Sciences, Magurele-Bucharest, Romania.

**Acknowledgements**

The research is supported by a grant from the "Research Center of the Center for Female Scientific and Medical Colleges", Deanship of Scientific Research, King Saud University. The authors are also thankful to visiting professor program at King Saud University for support.

**Competing interests**

The authors declare that they have no competing interests.

Received: 20 June 2016 Accepted: 15 August 2016

Published online: 27 August 2016

**References**

- Abbasbandy S, Otadi M (2006) Numerical solution of fuzzy polynomials by fuzzy neural network. *Appl Math Comput* 181:1084–1089
- Barthelmann V, Novak E, Ritter K (2000) High dimensional polynomial interpolation on sparse grids. *Adv Comput Math* 12:273–288
- Boffi D, Gastaldi L (2006) Interpolation estimates for edge finite elements and application to band gap computation. *Appl Numer Math* 56:1283–1292
- Chen Lh, Zhang Xy (2009) Application of artificial neural networks to classify water quality of the yellow river. *Fuzzy Inf Eng* 9:15–23
- Cybenko G (1989) Approximation by superpositions of a sigmoidal function. *Math Control Signals Syst* 2:303–314
- Guo B, Qin L (2009) Tactile sensor signal processing with artificial neural networks. *Fuzzy Inf Eng* 54:54–62
- Ishibuchi H, Kwon K, Tanaka H (1995) A learning of fuzzy neural networks with triangular fuzzy weights. *Fuzzy Sets Syst* 71:277–293
- Ito Y (2001) Independence of unscaled basis functions and finite mappings by neural networks. *Math Sci* 26:117–126
- Jafarian A, Measoomyinia S (2011) Solving fuzzy polynomials using neural nets with a new learning algorithm. *Appl Math Sci* 5:2295–2301
- Jafarian A, Jafari R (2012) Approximate solutions of dual fuzzy polynomials by feed-back neural networks. *J Soft Comput Appl*. doi:10.5899/2012/jsca-00005
- Jafarian A, Measoomyinia S (2012) Utilizing feed-back neural network approach for solving linear Fredholm integral equations system. *Appl Math Model*. doi:10.1016/j.apm
- Jafarian A, Jafari R, Khalili A, Baleanud D (2015a) Solving fully fuzzy polynomials using feedback neural networks. *Int J Comput Math* 92:742–755
- Jafarian A, Measoomy S, Abbasbandy S (2015b) Artificial neural networks based modeling for solving Volterra integral equations system. *Appl Soft Comput* 27:391–398
- Liu P (2000) Analyses of regular fuzzy neural networks for approximation capabilities. *Fuzzy Sets Syst* 114:329–338
- Llanas B, Sainz FJ (2006) Constructive approximate interpolation by neural networks. *J Comput Appl Math* 188:283–308
- Mastylo M (2010) Interpolation estimates for entropy numbers with applications to non-convex bodies. *J Approx Theory* 162:10–23
- Neidinger RD (2009) Multivariable interpolating polynomials in newton forms. In: Joint mathematics meetings, Washington, DC, pp 5–8
- Olver PJ (2006) On multivariate interpolation. *Stud Appl Math* 116:201–240
- Rajan D, Chaudhuri S (2001) Generalized interpolation and its application in super-resolution imaging. *Image Vis Comput* 19:957–969
- Schroeder H, Murthy VK, Krishnamurthy EV (1991) Systolic algorithm for polynomial interpolation and related problems. *Parallel Comput* 17:493–503
- Song Q, Zhao Z, Yang J (2013) Passivity and passification for stochastic TakagiSugeno fuzzy systems with mixed time-varying delays. *Neurocomputing* 122:330–337
- Sontag ED (1992) Feedforward nets for interpolation and classification. *J Comput Syst Sci* 45:20–48
- Szabados J, Vertesi P (1990) Interpolation of functions. World Scientific, Singapore
- Tikhomirov VM (1990) Approximation theory, analysis II. In: Gamkrelidze RV (ed) *Encyclopaedia of mathematical sciences*, vol 14. Springer, Berlin
- Wai RJ, Lin YW (2013) Adaptive moving-target tracking control of a vision-based mobile robot via a dynamic petri recurrent fuzzy neural network. *IEEE Trans Fuzzy Syst* 21:688–701
- Xiu D, Hesthaven JS (2005) High-order collocation methods for differential equations with random inputs. *SIAM J Sci Comput* 27:18–39
- Zadeh LA (2005) Toward a generalized theory of uncertainty (GTU) an outline. *Inf Sci* 172:1–40