Available online at www.ispacs.com/cna
Volume 2012, Year 2012 Article ID cna-00128, 13 pages
doi:10.5899/2012/cna-00128
Research Article

Fuzzy Fredholm integro-differential equations with artificial neural networks

Maryam Mosleh[1] *, Mahmood Otadi[1]
(1) *Department of Mathematics, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran*

Abstract
In this paper, we use parametric form of fuzzy number, then feed-forward neural network is presented for obtaining approximate solution for fuzzy Fredholm integro-differential equation of the second kind. This paper presents a method based on neural networks and Newton-Cotes methods with positive coefficient. The ability of neural networks in function approximation is our main objective. The proposed method is illustrated by solving some numerical examples.
Keywords: Fuzzy integro-differential equations; Artificial neural networks

1 Introduction

The solutions of integral equations have a major role in the field of science and engineering. A physical even can be modelled by the differential equation, an integral equation. Since few of these equations cannot be solved explicitly, it is often necessary to resort to numerical techniques which are appropriate combinations of numerical integration and interpolation [12, 32]. There are several numerical methods for solving linear Volterra integral equation [18, 41] and system of nonlinear Volterra integral equations [14]. Kauthen in [28] used a collocation method to solve the Volterra- Fredholm integral equation numerically. Borzabadi and Fard in [16] obtained a numerical solution of nonlinear Fredholm integral equations of the second kind.

The concept of fuzzy numbers and fuzzy arithmetic operations were first introduced by Zadeh [43], Dubois and Prade [20]. We refer the reader to [26] for more information on fuzzy numbers and fuzzy arithmetic. The topics of fuzzy integral equations (FIE)

*Corresponding author. E-mail: mosleh@iaufb.ac.ir; Tel: +98 912 6076308

which growing interest for some time, in particular in relation to fuzzy control, have been rapidly developed in recent years. The fuzzy mapping function was introduced by Chang and Zadeh [17]. Later, Dubois and Prade [21] presented an elementary fuzzy calculus based on the extension principle also the concept of integration of fuzzy functions was first introduced by Dubois and Prade [21]. Babolian et al. and Abbasbandy et al. in [3, 11] obtained a numerical solution of linear Fredholm fuzzy integral equations of the second kind. Allahviranloo et al. in [6] presented a new method for solving fuzzy integro-differential equation under generalized differentiability.

Another group of researchers tried to extend some numerical methods to solve fuzzy differential equatins (FDEs) [1, 8, 38] such as Runge-Kutta method [2], Adomian method [10], predictor-corrector method and multi-step methods [7].

Fuzzy neural network have been extensively studied [15, 19] and recently, successfully used for solving fuzzy polynomial equation and systems of fuzzy polynomials [4, 5], approximate fuzzy coefficients of fuzzy regression models [33, 34, 35], approximate solution of fuzzy linear systems and fully fuzzy linear systems [36, 39]. Lagaris et al. in [31] used multilayer perceptron to estimate the solution of differential equation. Their neural network model was trained over an interval (over which the differential equation must be solved), so the inputs of the neural network model were the training points. The comparison of their method with the existing numerical method shows that their method was more accurate and the solution had also more generalizations. Recently Effati et al. in [22] and Mosleh and Otadi [37] used artificial neural networks to solve fuzzy ordinary differential equations. But in this paper, we extend the artificial neural networks to solve integro-differential equations. The ability of neural networks in function approximation is our main objective. In this paper, we present a novel and very simple numerical method based upon neural networks for solving fuzzy linear Fredholm integro-differential equations of the second kind

$$X'(s) = y(s) + \lambda \int_a^b k(s,t)X(t)dt.$$

2 Preliminaries

In this section the basic notations used in fuzzy operations are introduced. We start by defining the fuzzy number.

Definition 2.1. *[30] A fuzzy number is a fuzzy set $u : \mathbb{R}^1 \longrightarrow I = [0,1]$ such that*

i. u is upper semi-continuous;

ii. $u(x) = 0$ outside some interval $[a,d]$;

iii. There are real numbers b and c, $a \le b \le c \le d$, for which

1. $u(x)$ is monotonically increasing on $[a,b]$,

2. $u(x)$ is monotonically decreasing on $[c,d]$,

3. $u(x) = 1, b \le x \le c$.

The set of all the fuzzy numbers (as given in Definition (2.1) is denoted by E^1. An alternative definition which yields the same E^1 is given by Kaleva [27].

Definition 2.2. *A fuzzy number u is a pair $(\underline{u}, \overline{u})$ of functions $\underline{u}(r)$ and $\overline{u}(r)$, $0 \le r \le 1$, which satisfy the following requirements:*

i. $\underline{u}(r)$ is a bounded monotonically increasing, left continuous function on $(0,1]$ and right continuous at 0;

ii. $\overline{u}(r)$ *is a bounded monotonically decreasing, left continuous function on* $(0, 1]$ *and right continuous at* 0*;*
iii. $\underline{u}(r) \leq \overline{u}(r), 0 \leq r \leq 1$.

A crisp number r is simply represented by $\underline{u}(\alpha) = \overline{u}(\alpha) = r, 0 \ \leq \alpha \leq 1$. The set of all the fuzzy numbers is denoted by E^1. This fuzzy number space as shown in [42], can be embedded into the Banach space $B = \overline{C}[0, 1] \times \overline{C}[0, 1]$.

Definition 2.3. *[27] For arbitrary* $u = (\underline{u}(r), \overline{u}(r)), v = (\underline{v}(r), \overline{v}(r))$*, we say that* $u = v$ *if and only if* $\underline{u} = \underline{v}$ *and* $\overline{u} = \overline{v}$*.*

For arbitrary $u = (\underline{u}(r), \overline{u}(r)), v = (\underline{v}(r), \overline{v}(r))$ *and* $k \in \mathbb{R}$ *we define addition and multiplication by* k *as*

$$\underline{(u + v)}(r) = (\underline{u}(r) + \underline{v}(r)),$$

$$\overline{(u + v)}(r) = (\overline{u}(r) + \overline{v}(r)),$$

$$\underline{ku}(r) = k\underline{u}(r), \overline{ku}(r) = k\overline{u}(r), \ if \ k \geq 0,$$

$$\underline{ku}(r) = k\overline{u}(r), \overline{ku}(r) = k\underline{u}(r), \ if \ k < 0.$$

Definition 2.4. *[24] For arbitrary fuzzy numbers* u, v*, we use the distance*

$$D(u, v) = sup_{0 \leq r \leq 1} max\{|\overline{u}(r) - \overline{v}(r)|, |\underline{u}(r) - \underline{v}(r)|\}$$

and it is shown that (E^1, D) *is a complete metric space [40].*

Definition 2.5. *[23, 24] Let* $f : [a, b] \longrightarrow E^1$*, for each partition* $P = \{t_0, t_1, \ldots, t_n\}$ *of* $[a, b]$ *and for arbitrary* $\xi_i \in [t_{i-1}, t_i], 1 \leq i \leq n$ *suppose*

$$R_p = \textstyle\sum_{i=1}^{n} f(\xi_i)(t_i - t_{i-1}),$$

$$\Delta := max\{|t_i - t_{i-1}|, i = 1, 2, \ldots, n\}.$$

The definite integral of $f(t)$ *over* $[a, b]$ *is*

$$\int_a^b f(t)dt = lim_{\Delta \longrightarrow 0} R_p$$

provided that this limit exists in the metric D*.*

If the fuzzy function $f(t)$ *is continuous in the metric* D*, its definite integral exists [24] and also,*

$$\underline{(\textstyle\int_a^b f(t; r)dt)} = \textstyle\int_a^b \underline{f}(t; r)dt,$$

$$\overline{(\textstyle\int_a^b f(t; r)dt)} = \textstyle\int_a^b \overline{f}(t; r)dt.$$

Definition 2.6. *Let* $u, v \in E^1$*. If there exists* $w \in E^1$ *such that* $u = v + w$ *then* w *is called the H-difference of* u, v *and it is denoted by* $u - v$*.*

Definition 2.7. *A function $f:(a,b)\longrightarrow E^1$ is called H-differentiable at $\hat{t}\in(a,b)$ if, for $h>0$ sufficiently small, there exist the H-differences $f(\hat{t}+h)-f(\hat{t}), f(\hat{t})-f(\hat{t}-h)$, and an element $f'(\hat{t})\in E^1$ such that:*

$$lim_{h\longrightarrow 0^+}D(\frac{f(\hat{t}+h)-f(\hat{t})}{h},f'(\hat{t}))=lim_{h\longrightarrow 0^+}D(\frac{f(\hat{t})-f(\hat{t}-h)}{h},f'(\hat{t}))=0.$$

Then $f'(\hat{t})$ is called the fuzzy derivative of f at $\hat{t}$.

3 Fuzzy integro-differential equation

The linear Fredholm integro-differential equations [25]

$$X'(s)=y(s)+\lambda\int_a^b k(s,t)X(t)dt,\quad X(s_0)=X_0, \tag{3.1}$$

where $\lambda>0$, k is an arbitrary given kernel function over the square $a\le s,t\le b$ and $y(s)$ is a given function of $s\in[a,b]$. If X is a fuzzy function, $y(s)$ is a given fuzzy function of $s\in[a,b]$ and X' is the fuzzy derivative (according to Definition (2.7)) of X, this equation may only possess fuzzy solution. Sufficient for the existence equation of the second kind, are given in [13].

Let $X(s)=(\underline{X}(s;r),\overline{X}(s;r))$ is a fuzzy solution of Eq.(3.1), therefore by Definition (2.3), Definition (2.5) and Definition (2.7) we have the equivalent system

$$\begin{aligned}&\underline{X}'(s)=\underline{y}(s)+\lambda\textstyle\int_a^b \underline{k(s,t)X(t)}dt,\quad \underline{X}(s_0)=\underline{X_0},\\ &\overline{X}'(s)=\overline{y}(s)+\lambda\textstyle\int_a^b \overline{k(s,t)X(t)}dt,\quad \overline{X}(s_0)=\overline{X_0}\end{aligned} \tag{3.2}$$

which possesses a unique solution $(\underline{X},\overline{X})\in B$ which is a fuzzy function, i.e. for each s, the pair $(\underline{X}(s;r),\overline{X}(s;r))$ is a fuzzy number, therefore each solution of Eq.(3.1) is a solution of system (3.2) and conversely also Eq.(3.1) and system (3.2) are equivalent.

The parametric form of Eqs.(3.2) is given by

$$\begin{aligned}&\underline{X}'(s,r)=\underline{y}(s,r)+\lambda\textstyle\int_a^b \underline{k(s,t)X(t,r)}dt,\quad \underline{X}(s_0)=\underline{X_0}(r),\\ &\overline{X}'(s,r)=\overline{y}(s,r)+\lambda\textstyle\int_a^b \overline{k(s,t)X(t,r)}dt,\quad \overline{X}(s_0)=\overline{X_0}(r)\end{aligned} \tag{3.3}$$

for $r\in[0,1]$. suppose $k(s,t)$ be continuous in $a\le s\le b$ and for fix t, $k(s,t)$ changes its sing in finite points as s_i where $x_i\in[a,s_1]$. For example, let $k(s,t)$ be nonnegative over $[a,s_1]$ and negative over $[s_1,b]$, therefore we have

$$\underline{X}'(s,r)=\underline{y}(s,r)+\lambda\textstyle\int_a^{s_1}k(s,t)\underline{X(t,r)}dt+\lambda\int_{s_1}^b k(s,t)\overline{X(t,r)}dt,\quad \underline{X}(s_0)=\underline{X_0}(r),$$

$$\overline{X}'(s,r)=\overline{y}(s,r)+\lambda\textstyle\int_a^{s_1}k(s,t)\overline{X(t,r)}dt+\lambda\int_{s_1}^b k(s,t)\underline{X(t,r)}dt,\quad \overline{X}(s_0)=\overline{X_0}(r).$$

In most cases, however, analytical solution to Eq.(3.3) may not be found and a numerical approach must be considered.

4 Function approximation

The use of neural networks provides solutions with very good generalizability (such as differentiability). On the other hand, an important feature of multi-layer perceptrons is their utility to approximate functions, which leads to a wide applicability in most problems.

In this paper, the function approximation capabilities of feed-forward neural networks is used by expressing the trial solutions for a system (3.3) as the sum of two terms. The first term satisfies the intial conditions and contains no adjustable parameters. The second term involves a feed-forward neural network to be trained so as to satisfy the integro-differential equations. Since it is known that a multilayer perceptron with one hidden layer can approximate any function to arbitrary accuracy, the multilayer perceptron is used as the type of the network architecture.

If $\underline{X}_T(s,r,\underline{p})$ is a trial solution for the first equation in system (3.3) and $\overline{X}_T(s,r,\overline{p})$ is a trial solution for the second equation in the system (3.3) where $\underline{p}$ and $\overline{p}$ are adjustable parameters, (indeed $\underline{X}_T(s,r,\underline{p})$ and $\overline{X}_T(s,r,\overline{p})$ are approximations of $\underline{X}(s,r)$ and $\overline{X}(s,r)$ respectively) thus the problem of finding the approximated solutions for (3.3) over some collocation point in $[a,b]$ is equivalent to calculate the functionals $\underline{X}_T$ and $\overline{X}_T$ that satisfies the following constrained optimization problem [29]:

$$\begin{cases} min_{\breve{p}} \sum_{i=1}^{M}\{(\underline{X}'_T(s_i,r,\underline{p}) - \underline{y}(s_i,r) - \underline{F}(s_i,r,\breve{p}))^2 + \\ \qquad (\overline{X}'_T(s_i,r,\overline{p}) - \overline{y}(s_i,r) - \overline{F}(s_i,r,\breve{p}))^2\}, \\ \underline{X}_T(s_0,r,\underline{p}) = \underline{X}_0(r), \quad \overline{X}_T(s_0,r,\overline{p}) = \overline{X}_0(r) \end{cases} \tag{4.4}$$

where $\breve{p} = (\underline{p}, \overline{p})$ contain all adjustable parameters (weights of input and output layers and biases) and

$$\underline{F}(s,r,\breve{p}) = \lambda \int_a^b \underline{k(s,t)X_T(t,r,\breve{p})}dt,$$
$$\overline{F}(s,r,\breve{p}) = \lambda \int_a^b \overline{k(s,t)X_T(t,r,\breve{p})}dt.$$

In general we cannot be able to carry out analytically the integrations, involved. In this case we naturally turn to numerical quadrature. We introduce a quadrature rule R for the interval $[a,b]$ with positive weights w_j and N nodes t_j, i.e.,

$$Rf = \sum_{j=1}^{N} w_j f(t_j) = If - Ef = \int_a^b f(t)dt - Ef,$$

where Ef is the error. If we first ignore the error of this quadrature rule then the Eq. (4.4) is replaced by the approximate equation

$$\begin{cases} min_{\breve{p}} \sum_{i=1}^{M}\{(\underline{X}'_T(s_i,r,\underline{p}) - \underline{y}(s_i,r) - \lambda \sum_{j=1}^{N} w_j \underline{k(s_i,t_j)X_T(t_j,r,\breve{p})})^2 + \\ \qquad (\overline{X}'_T(s_i,r,\overline{p}) - \overline{y}(s_i,r) - \lambda \sum_{j=1}^{N} w_j \overline{k(s_i,t_j)X_T(t_j,r,\breve{p})})^2\}, \\ \underline{X}_T(s_0,r,\underline{p}) = \underline{X}_0(r), \quad \overline{X}_T(s_0,r,\overline{p}) = \overline{X}_0(r). \end{cases} \tag{4.5}$$

Each trial solution $\underline{X}_T$ and $\overline{X}_T$ employs one feed-forward neural network for wich the corresponding networks are denoted by $\underline{N}$ and $\overline{N}$, with adjustable parameters $\underline{p}$ and $\overline{p}$, respectively. The related trial functions will be in the form [22]:

$$\begin{aligned} \underline{X}_T(s,r,\underline{p}) &= \underline{X}(s_0,r) + (s-s_0)\underline{N}(s,r,\underline{p}), \\ \overline{X}_T(s,r,\overline{p}) &= \overline{X}(s_0,r) + (s-s_0)\overline{N}(s,r,\overline{p}), \end{aligned} \tag{4.6}$$

where $\underline{N}$ and $\overline{N}$ are single-output feed-forward neural networks with adjustable parameters $\underline{p}$ and $\overline{p}$, respectively. Here s and r are the network inputs. This solutions by intention satisfies the initial condition in (4.6). According to (4.6) it is straight forward to show that:

$$\begin{aligned} \underline{X}'_T(s,r,\underline{p}) &= \underline{N}(s,r,\underline{p}) + (s-s_0)\frac{\partial \underline{N}}{\partial s}, \\ \overline{X}'_T(s,r,\overline{p}) &= \overline{N}(s,r,\overline{p}) + (s-s_0)\frac{\partial \overline{N}}{\partial s}. \end{aligned} \tag{4.7}$$

Now consider a multilayer perceptron having one hidden layer with H sigmoid units and a linear output unit (Fig.1, Fig. 2).

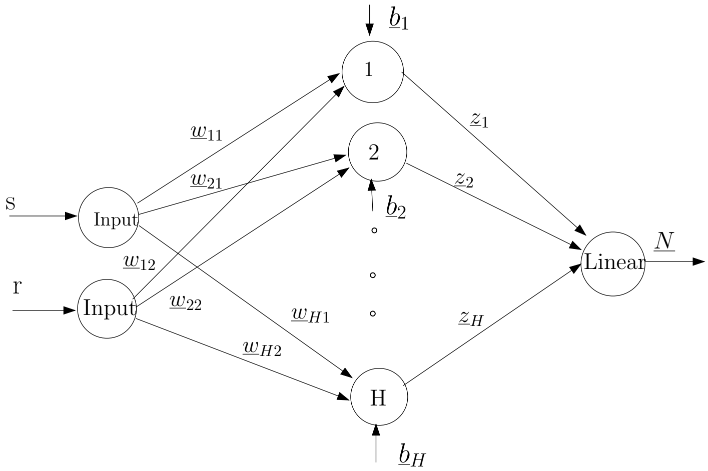

Fig. 1. Three layered perceptron with two input and $\underline{N}$ output.

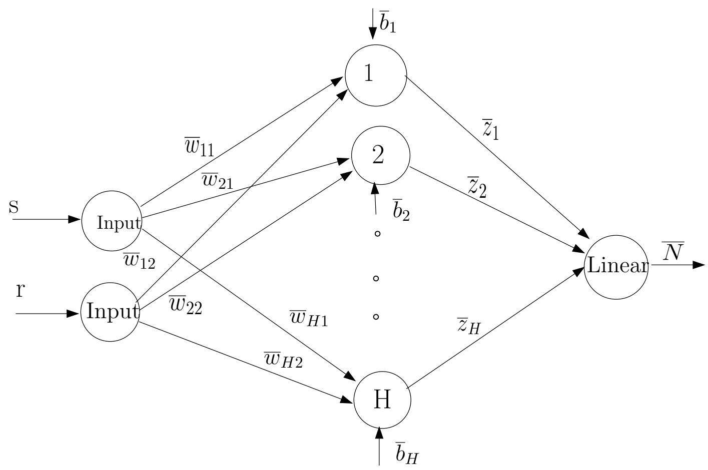

Fig. 2. Three layered perceptron with two input and $\overline{N}$ output.

Here we have:

$$\begin{aligned} \underline{N} &= \textstyle\sum_{i=1}^{H} \underline{v}_i \sigma(\underline{z_i}), \quad \underline{z_i} = \underline{w}_{i1}s + \underline{w}_{i2}r + \underline{b}_i, \\ \overline{N} &= \textstyle\sum_{i=1}^{H} \overline{v}_i \sigma(\overline{z_i}), \quad \overline{z_i} = \overline{w}_{i1}s + \overline{w}_{i2}r + \overline{b}_i, \end{aligned} \tag{4.8}$$

where $\sigma(z)$ is the sigmoid transfer function. The following is obtained:

$$\begin{aligned} \frac{\partial \underline{N}}{\partial s} &= \textstyle\sum_{i=1}^{H} \underline{v}_i \underline{w}_{i1} \sigma'(\underline{z_i}), \\ \frac{\partial \overline{N}}{\partial s} &= \textstyle\sum_{i=1}^{H} \overline{v}_i \overline{w}_{i1} \sigma'(\overline{z_i}), \end{aligned}$$

where $\sigma'(\bar{z}_i)$ is the first derivative of the sigmoid function. Also there are many choices for the sigmoid function, here we choose $\sigma(z) = 1/(1+e^{-z})$ since it is possible to derive all the derivatives of $\sigma(z)$ in terms of the sigmoid function itself. i.e.

$$\sigma'(z) = -\sigma^2(z) + \sigma(z).$$

5 Example

To illustrate the technique proposed in this paper, consider the following example. For each fuzzy numbers, we use $r = 0, 0.1, \ldots, 1$, where we calculate the accuracy of the method by Eq. (4.5). In the computer simulation of this section, we use the $H = 10$ sigmoid units in the hidden layer.

Example 5.1. *Consider the following fuzzy linear Fredholm integro-differential equation*

$$\begin{gathered} X'(s) = (0.5+0.5r, 2-r)(e^s - s) + \int_0^1 tsX(t)dt, \\ X(0) = (0.5+0.5r, 2-r); \quad 0 \le r \le 1, \quad 0 \le s, t \le 1. \end{gathered}$$

The exact solution in this case is given by

$$X = (0.5+0.5r, 2-r)e^s.$$

The trial functions for this problem are

$$\underline{X}_T(s;r) = (0.5+0.5r) + s\sum_{i=1}^{H} \frac{\underline{v}_i}{1+e^{-\underline{w}_{i1}s-\underline{w}_{i2}r-\underline{b}_i}},$$

$$\overline{X}_T(s;r) = (2-r) + s\sum_{i=1}^{H} \frac{\overline{v}_i}{1+e^{-\overline{w}_{i1}s-\overline{w}_{i2}r-\overline{b}_i}}.$$

The exact and obtained solution of fuzzy linear Fredholm integro-differential equation in this example at $s = 1$ are shown in Figure 3, also the error by Eq. (4.5) is $1.124331e^{-4}$.

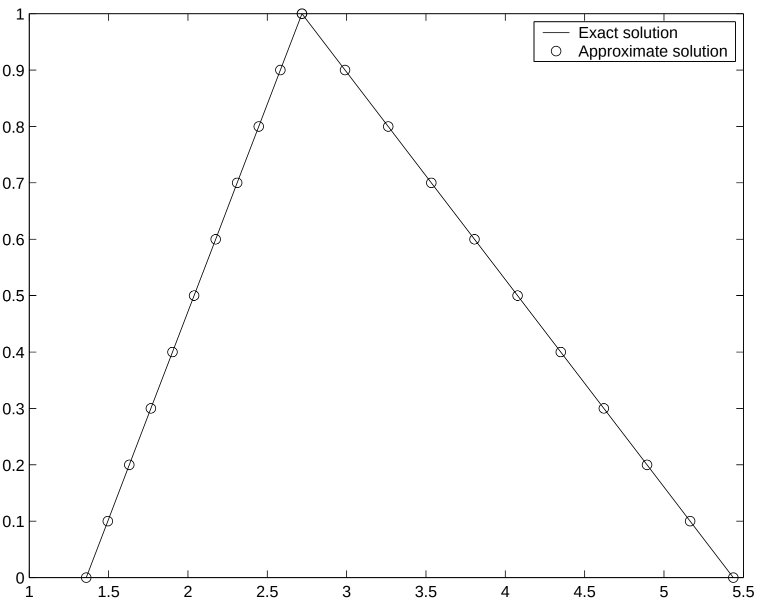

Fig. 3. The exact and approximate solution for example 5.1.

Figs. 4-7 show the convergence property of the computed values of the weights.

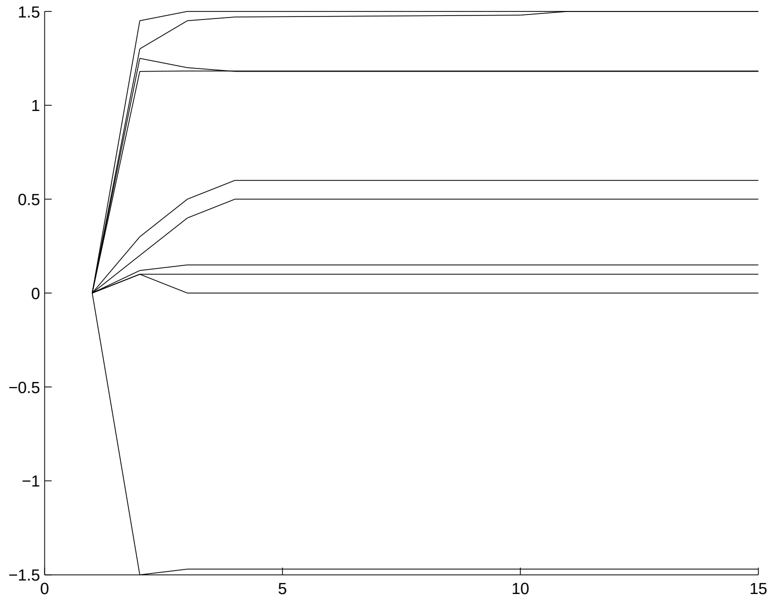

Fig. 4. Convergence of the weights $\underline{w}_{i1}$ for example 5.1.

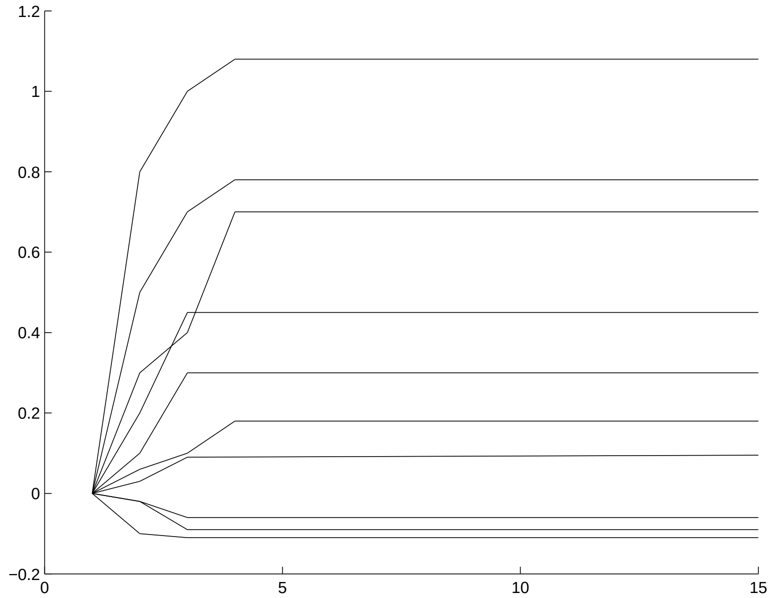

Fig. 5. Convergence of the weights $\underline{w}_{i2}$ for example 5.1.

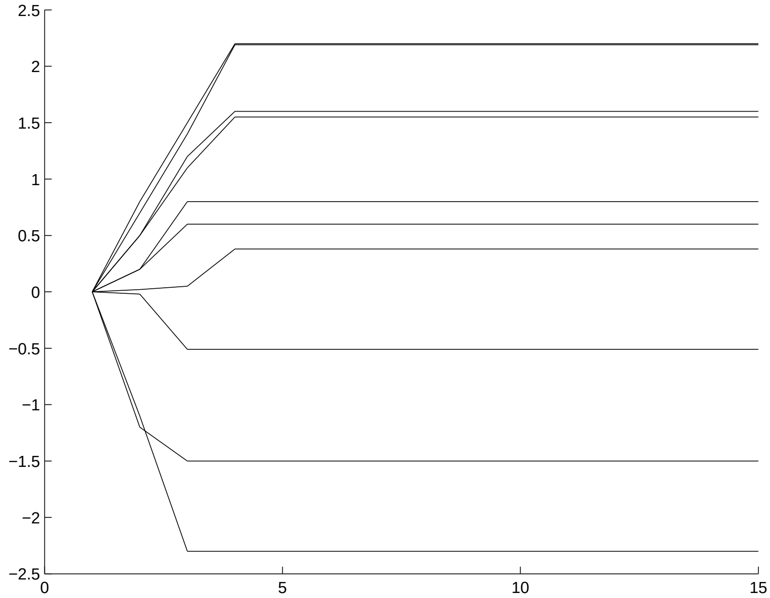

Fig. 6. Convergence of the weights $\underline{v}_i$ for example 5.1.

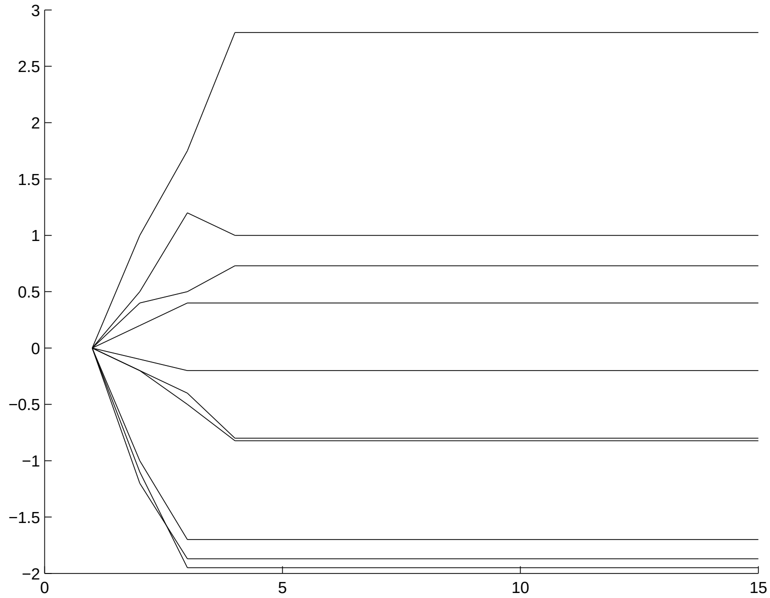

Fig. 7. Convergence of the weights $\underline{b}_i$ for example 5.1.

6 Conclusion

Solving fuzzy integro-differential equation (FIDE) by using universal approximators (UA), that is, neural network model (NNM) is presented in this paper.

In this paper, the original fuzzy integro-differential equation is replaced by two parametric linear Fredholm integro-differential equations which are then solved numerically using UAM. The main reason for using neural networks was their applicability in function approximation. Our computer simulation in this paper were performed for three-layer feedforward neural networks. Since we had good simulation result even from three-layer neural networks, we do not think that the extension of our NNM to neural networks with more than three layers is an attractive research direction.

References

[1] S. Abbasbandy, T. A. Viranloo, Ó. López, J. J. Nieto, Numerical methods for fuzzy differential inclusions, Computers and Mathematics with Applications, 48 (2004) 1633-1641.
http://dx.doi.org/10.1016/j.camwa.2004.03.009

[2] S. Abbasbandy and T. Allahviranloo, Numerical solution of fuzzy differential equations by Runge-Kutta method, Nonlinear Studies, 11 (1) (2004) 117-129.

[3] S. Abbasbandy, E. Babolian, M. Alavi, Numerical method for solving linear Fredholm fuzzy integral equations of the second kind, Chaos Solitons & Fractals, 31 (2007) 138-146.

[4] S. Abbasbandy and M. Otadi, Numerical solution of fuzzy polynomials by fuzzy neural network, Applied Mathematics and Computation, 181 (2006) 1084-1089.
http://dx.doi.org/10.1016/j.amc.2006.01.073

[5] S. Abbasbandy, M. Otadi and M. Mosleh, Numerical solution of a system of fuzzy polynomials by fuzzy neural network, Information Sciences, 178 (2008) 1948-1960.
http://dx.doi.org/10.1016/j.ins.2007.11.026

[6] T. Allahviranloo, S. Abbasbandy, O. Sedaghatfar and P. Darabi, A new method for solving fuzzy integro-differential equation under generalized differentiability, Neural Computing and Applications, 21 (2012) 191-196.
http://dx.doi.org/10.1007/s00521-011-0759-3

[7] T. Allahviranloo, N. Ahmadi and E. Ahmadi, Numerical solution of fuzzy differential equations by predictor-corrector method, Informations Sciences, 177 (2007) 1633-1647.
http://dx.doi.org/10.1016/j.ins.2006.09.015

[8] T. Allahviranloo, N.A. Kiani and M. Barkhordari, Toward the existence and uniqueness of solutions of second-order fuzzy differential equations, Informations Sciences, 177 (2009) 1207-1215.
http://dx.doi.org/10.1016/j.ins.2008.11.004

[9] T. Allahviranloo, M. Khezerloo, O. Sedaghatfar and S. Salahshour, Toward the existence and uniqueness of solutions of second-order fuzzy volterra integro-differential equations with fuzzy kernel, Neural Computing and Applications.
http://dx.doi.org/10.1007/s00521-012-0849-x

[10] E. Babolian, H. Sadeghi and Sh. Javadi, Numerically solution of fuzzy differential equations by Adomian method, Applied Mathematics and Computation, 149 (2004) 547-557.
http://dx.doi.org/10.1016/S0096-3003(03)00160-7

[11] E. Babolian, H.S. Goghary, S. Abbasbandy, Numerical solution of linear Fredholm fuzzy integral equations of the second kind by Adomian method, Applied Mathematics and Computation, 161 (2005) 733-744.
http://dx.doi.org/10.1016/j.amc.2003.12.071

[12] C.T.H. Baker, A perspective on the numerical treatment of Volterra equations, J. Comput. Appl. Math., 125 (2000) 217-249.
http://dx.doi.org/10.1016/S0377-0427(00)00470-2

[13] P. Balasubramaniam, S. Muralisankar, Existence and uniqueness of fuzzy solution for the nonlinear fuzzy integro-differential equations, Applied mathematics letters, 14 (2001) 455-462.
http://dx.doi.org/10.1016/S0893-9659(00)00177-4

[14] M.I. Berenguer, D.Gamez, A.I. Garralda-Guillem, M. Ruiz Galan, M.C. Serrano Perez, Biorthogonal systems for solving Volterra integral equation systems of the second kind, J. Comput. Appl. Math., 235 (2011) 1875-1883.
http://dx.doi.org/10.1016/j.cam.2010.07.011

[15] J.F. Bernard, Use of rule-based system for process control, IEEE Contr. System Mag., 8 (1988) 3-13.
http://dx.doi.org/10.1109/37.7735

[16] A.H. Borzabadi and O.S. Fard, A numerical scheme for a class of nonlinear Fredholm integral equations of the second kind, Journal of Computational and Applied Mathematics, 232 (2009) 449-454.
http://dx.doi.org/10.1016/j.cam.2009.06.038

[17] S.S.L. Chang, L. Zadeh, On fuzzy mapping and control, IEEE Trans. System Man Cybernet, 2 (1972) 30-34.

[18] Y. Chen, T. Tang, Spectral methods for weakly singular Volterra integral equations with smooth solutions, J. Comput. Appl. Math., 233 (2009) 938-950.
http://dx.doi.org/10.1016/j.cam.2009.08.057

[19] Y.C. Chen, C.C. Teng, A model reference control structure using a fuzzy neural network, Fuzzy Sets and Systems, 73 (1995) 291-312.
http://dx.doi.org/10.1016/0165-0114(94)00319-3

[20] D. Dubois and H. Prade, Operations on fuzzy numbers, J. Systems Sci. 9 (1978) 613-626.
http://dx.doi.org/10.1080/00207727808941724

[21] D. Dubois, H. Prade, Towards fuzzdifferential calculus, Fuzzy Sets Systems, 8 (1982) 1-7.
http://dx.doi.org/10.1016/0165-0114(82)90025-2

[22] S. Effati and M. Pakdaman, Artificial neural network approach for solving fuzzy differential equations, Information Sciences, 180 (2010) 1434-1457.
http://dx.doi.org/10.1016/j.ins.2009.12.016

[23] M. Friedman, M. Ma, A. Kandel, Numerical solutions of fuzzy differential and integral equations, Fuzzy Sets and Systems 106 (1999) 35-48.
http://dx.doi.org/10.1016/S0165-0114(98)00355-8

[24] R. Goetschel, W. Vaxman, Elementary calculus, Fuzzy sets Syst., 18 (1986) 31-43.
http://dx.doi.org/10.1016/0165-0114(86)90026-6

[25] H. Hochstadt, Integral equations, New York: Wiley; 1973.

[26] A. Kaufmann and M.M. Gupta, Introduction Fuzzy Arithmetic, Van Nostrand Reinhold, New York, 1985.

[27] O. Kaleva, Fuzzy differential equations, Fuzzy Sets Syst. 24 (1987) 301-317.
http://dx.doi.org/10.1016/0165-0114(87)90029-7

[28] J.P. Kauthen, Continuous time collocation method for Volterra-Fredholm integral equations, Numer. Math. 56 (1989) 409-424.
http://dx.doi.org/10.1007/BF01396646

[29] D. R. Kincaid, E.W. Cheney, Numerical analysis: Mathematics of scientific comuting, third ed., Brooks/Cole, Pacific Grove, CA, 2002.

[30] G.J. Klir, U.S. Clair, B. Yuan, Fuzzy set theory: foundations and applications, Prentice-Hall Inc.;1997.

[31] I. E. Lagaris and A. Likas, Artificial neural networks for solving ordinary and partial differential equations, IEEE Transactions on Neural Networks 9 (5) (1998), September.

[32] P. Linz, Analytical and numerical methods for Volterra equations, SIAM, Philadelphia, PA, 1985.
http://dx.doi.org/10.1137/1.9781611970852

[33] M. Mosleh, M. Otadi and S. Abbasbandy, Evaluation of fuzzy regression models by fuzzy neural network, Journal of Computational and Applied Mathematics, 234 (2010) 825-834.
http://dx.doi.org/10.1016/j.cam.2010.01.046

[34] M. Mosleh, M. Otadi and S. Abbasbandy, Fuzzy polynomial regression with fuzzy neural networks, Applied Mathematical Modelling, 35 (2011) 5400-5412.
http://dx.doi.org/10.1016/j.apm.2011.04.039

[35] M. Mosleh, T. Allahviranloo and M. Otadi, Evaluation of fully fuzzy regression models by fuzzy neural network, Neural Comput and Applications, 21 (2012) 105-112.
http://dx.doi.org/10.1007/s00521-011-0698-z

[36] M. Otadi and M. Mosleh, Simulation and evaluation of dual fully fuzzy linear systems by fuzzy neural network, Applied Mathematical Modelling, 35 (2011) 5026-5039.
http://dx.doi.org/10.1016/j.apm.2011.04.008

[37] M. Mosleh and M. Otadi, Simulation and evaluation of fuzzy differential equations by fuzzy neural network, Applied Soft Computing, 12 (2012) 2817-2827.
http://dx.doi.org/10.1016/j.asoc.2012.03.041

[38] M. Mosleh and M. Otadi, Minimal solution of fuzzy linear system of differential equations, Neural Computing and Applications, 21 (2012) 329-336.
http://dx.doi.org/10.1007/s00521-012-0913-6

[39] M. Otadi, M. Mosleh and S. Abbasbandy, Numerical solution of fully fuzzy linear systems by fuzzy neural network, Soft Computing, 15 (2011) 1513-1522.
http://dx.doi.org/10.1007/s00500-010-0685-9

[40] M.L. Puri, D. Ralescu, Fuzzy random variables, J Math. Anal. Appl., 114 (1986) 409-422.
http://dx.doi.org/10.1016/0022-247X(86)90093-4

[41] H.H. Sorkun, S. Yalcinbas, Approximate solutions of linear Volterra integral equation systems with variable coefficients, Applied Mathematical Modelling, 34 (2010) 3451-3464.
http://dx.doi.org/10.1016/j.apm.2010.02.034

[42] Wu Congxin, Ma Ming, On embedding problem of fuzzy number spaces, Part 1, Fuzzy Sets and Systems, 44 (1991) 33-38.
http://dx.doi.org/10.1016/0165-0114(91)90030-T

[43] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Inform. Sci. 8 (1975) 199-249.
http://dx.doi.org/10.1016/0020-0255(75)90036-5