

Universal approximation method for the solution of integral equations

Mahmood Otadi¹ · Maryam Mosleh¹

Received: 18 July 2016 / Accepted: 8 February 2017 / Published online: 16 February 2017
© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract In this paper, we present a novel approach for approximating Hammerstein–Volterra delay integral equations (HVDIEs) by applying the universal approximation method through an artificial intelligence utility in a simple way. In this paper, neural network model (NNM) is applied as universal approximators for any nonlinear continuous functions. Here, neural network is considered as a part of large field called neural computing or soft computing. With this capability, the solution of Hammerstein–Volterra delay integral equation can be approximated by the appropriate NNM within an arbitrary accuracy.

Keywords Hammerstein–Volterra delay integral equation · Universal approximation · Neural network model · Numerical method · Mathematical model in epidemiology

Introduction

Proper design for engineering applications requires detailed information of the system-property distributions such as temperature, velocity, density, etc., in space and time domain. This information can be obtained by either experimental measurement or computational simulation. Although experimental measurement is reliable, it needs a lot of labor efforts and time. Therefore, the computational

simulation has become a more and more popular method as a design tool, since it needs only a fast computer with a large memory [45].

The solutions of integral equations have a major role in the field of science and engineering. The theory and application of integral equation are an important subject within applied mathematics. Integral equations are used as mathematical models for many physical situations, and integral equations also occur as reformulations of other mathematical problems, such as partial differential equations and ordinary differential equations. A physical even can be modeled 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 [3, 4, 6, 7, 30]. There are several numerical methods for solving linear Volterra integral equation [16, 49] and system of nonlinear Volterra integral equations [9, 51]. Biazar [12] presented differential transform method for solving systems of integral equations. Kauthen in [25] used a collocation method to solve the Volterra–Fredholm integral equation numerically. Borzabadi and Fard in [14] obtained a numerical solution of nonlinear Fredholm integral equations of the second kind.

In recent years, meshless methods have been developed as alternative numerical approaches in effort to eliminate known shortcomings of the mesh-based methods [8]. The main advantage of these methods is to approximate the unknowns by a linear combination of shape functions. Shape functions are based on a set of nodes and a certain weight function with a local support associated with each of these nodes. Therefore, they can solve many engineering problems that are not suited to the conventional computational methods [17, 47, 53]. Bhrawy et al. [10] reported a new spectral collocation technique for solving second kind

✉ Mahmood Otadi
mahmoodotadi@yahoo.com

Maryam Mosleh
maryamosleh79@yahoo.com

¹ Department of Mathematics, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran

Fredholm integral equations (FIEs). They developed a collocation scheme to approximate FIEs by means of the shifted Legendre–Gauss–Lobatto collocation (SL–GL–C) method. Then, they developed an efficient direct solver for solving numerically the high-order linear Fredholm integro-differential equations (FIDEs) with piecewise intervals under the initial-boundary conditions [11]. Maleknejad et al. [33] introduced an approach for obtaining the numerical solution of the nonlinear Volterra–Fredholm integro-differential (NVFID) equations using hybrid Legendre polynomials and block-pulse functions. These hybrid functions and their operational matrices are used for representing matrix form of these equations. Hashemizadeh and Rostami [20] obtained numerical solution of Hammerstein integral equations of mixed type by means of Sinc collocation method is presented. This proposed approximation reduces these kinds of nonlinear Hammerstein integral equations of mixed type to a nonlinear system of algebraic equations.

FNN systems are hybrid systems that combine the theories of fuzzy logic and neural networks. Designing the FNN system based on the input–output data is a very important problem. Several authors investigated FNN, to compute crisp and even fuzzy information with neural network (NN). There existed only a few approaches to learning algorithms for FNN when Ishibuchi et al. presented two NNs which can be trained with interval vectors and with vectors of fuzzy numbers. In both networks, Ishibuchi et al. used crisp weights. For these networks, they presented a backpropagation based on learning algorithm. In a later paper [23], Ishibuchi et al. developed an FNN with symmetric triangular fuzzy numbers as weights. For this NN, they evolved a learning algorithm in which the backpropagation algorithm is used to compute the new lower and upper limits of the support of the weights. The modal value of the new fuzzy weight is calculated as an average of the newly computed limits. Recently, FNN successfully was used for solving fuzzy polynomial equation and systems of fuzzy polynomials [1, 2], approximate fuzzy coefficients of fuzzy regression models [37, 38, 40, 41], approximate solution of fuzzy linear systems and fully fuzzy linear systems [42, 43], and fuzzy differential equations [34–36, 44].

In this work, we propose a new solution method for the approximate solution of integral equations using innovative mathematical tools and neural-like systems of computation. This hybrid method can result in improved numerical methods for solving integral equations. In this proposed method, NNM is applied as universal approximator. Neural computation research, together with related areas in approximation theory, has developed powerful methods for approximating continuous and integrable functions on compact subsets. In such schemes, function approximation

capabilities critically depend on the activation function nature of the hidden layer. Ito and Saito [24] proved that if the activation function is continuous and nondecreasing sigmoidal function, then the interpolation can be made with inner weights.

Neural network model

Artificial neural networks are an exciting form of artificial intelligence, which mimic the learning process of the human brain to extract patterns from historical data [2, 48]. For many years, this technology has been successfully applied to a wide variety of real-world applications [46]. Simple perceptrons need a teacher to tell the network what the desired output should be. These are supervised networks. In an unsupervised net, the network adapts purely in response to its inputs. Such networks can learn to pick out structure in their input. Figure 1 shows typical three-layered perceptron as a basic structural architecture with an input layer, a single hidden layer, and an output layer [19]. Here, the dimension of NNM is denoted by the number of neurons in each layer, that is $N \times M \times S$ NNM, where N , M , and S are the number of neurons in the input layer, the hidden layer, and the output layer, respectively. The architecture of the model shows how NNM transforms the N inputs $(t_1, \dots, t_i, \dots, t_N)$ into the s outputs $(x_1, \dots, x_k, \dots, x_S)$ throughout the m hidden neurons $(z_1, \dots, z_j, \dots, z_M)$, where the cycles represent the neurons in each layer. Let b_j be the bias for neuron z_j , c_k be the bias for neuron x_k , w_{ji} be the weight connecting neuron x_i to neuron z_j , and w_{kj} be the weight connecting neuron z_j to neuron x_k . Then, the output of NNM $f : \mathbb{R}^N \rightarrow \mathbb{R}^S$ can be determined as follows:

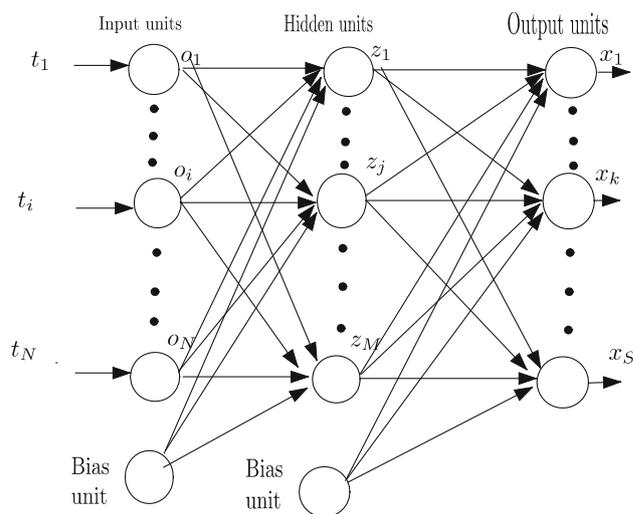


Fig. 1 Multiple layer feed-forward NNM



$$x_k = f_k \left(\sum_{j=1}^M w_{kj} z_j + c_k \right), \tag{1}$$

with

$$z_j = f_j \left(\sum_{i=1}^N w_{ji} t_i + b_j \right), \tag{2}$$

where f_k and f_j are the activation functions which are linear, piecewise linear, hard limiter, unipolar sigmoidal, and bipolar sigmoidal functions. The usual choices of the activation function [21, 28, 45] are the unipolar sigmoidal of the form:

$$f(\theta) = \frac{1}{1 + \exp(-\theta)}. \tag{3}$$

Multi-layered perceptrons with more than three layers, which use more hidden layers [21, 26]. Multi-layered perceptrons correspond the input units to the output units by a specific nonlinear mapping [50]. The most important application of multi-layered perceptrons is their ability in function approximation [15]. From Kolmogorov existence theorem, we know that a three-layered perceptron with $n(2n + 1)$ nodes can compute any continuous function of n variables [22, 31]. The accuracy of the approximation depends on the number of neurons in the hidden layer and does not depend on the number of the hidden layers [27].

Hammerstein–Volterra delay integral equations

The present paper deals with the investigation of the Volterra integral equation having a constant delay:

$$x(t) = g(t) + \lambda \int_{t-\tau}^t H(t, s) \cdot x(s) ds, \quad t \in [0, T], \tag{4}$$

with the initial condition

$$x(t) = \psi(t), \quad t \in [-\tau, 0], \tag{5}$$

where $\tau > 0$ and $T > 0$ are such that $T = p \bullet \tau$ for given $p \in \mathbb{N}$, and g and ψ are known functions and $H : [0, T] \times [-\tau, T] \rightarrow \mathbb{R}$ is a weight function. This equation is a mathematical model for the spread of certain infectious diseases with a contact rate that varies seasonally. Here, $x(t)$ is the proportion of infectives in the population at time t , $\tau > 0$ is the length of time in which an individual remains infectious, and $H(t, s)x(s)$ is the proportion of new infectives per unit time [39]. Throughout this paper, we always assume that the solution of (4) and (5) exists and is unique. Now, we study the NNM to approximate the solution of Eqs. (4) and (5).

The integral Eq. (4) contains a constant delay and its variant is the generalization of an epidemic model (see

[18]), where $H(t, s) = P(t - s)$. First, we define the operator:

$$T_1(t, x(t)) = x(t) - g(t) - \lambda \int_{t-\tau}^t H(t, s) \cdot x(s) ds, \tag{6}$$

$$T_2(t, x(t)) = x(t) - \psi(t). \tag{7}$$

To obtain approximate solution $x_M(t, P)$, we solve unconstrained optimization problem that is simpler to deal with; we define the trial function to be in the following form:

$$x_M(t, P) = N_M(t, P), \tag{8}$$

where the term in the right-hand side is a feed-forward neural network consisting of an input t and P is the vector containing all the adjustable parameters of NNM [32]. This NNM with some weights and biases is considered, and we train to compute the approximate solution of HVDIEs.

Substituting (8) into Eqs. (4) and (5), we can obtain the expression:

$$T_1(t, x_M(t)) = x_M(t) - g(t) - \lambda \int_{t-\tau}^t H(t, s) \cdot x_M(s) ds, \tag{9}$$

$$t \in [0, T],$$

$$T_2(t, x_M(t)) = x_M(t) - \Phi(t), \quad t \in [-\tau, 0]. \tag{10}$$

For any $t \in [-\tau, 0]$, $R_{M_2}(t) = T_2(t, x_M(t)) - T_2(t, x(t))$ and for any $t \in [0, T]$, $R_{M_1}(t) = T_1(t, x_M(t)) - T_1(t, x(t))$ is called the remaining items of Eqs. (4) and (5), where M is the numbers of neurons or hidden layers for NNM; also in addition, we have:

$$R_{M_1}(t) = (x_M(t) - x(t)) - \lambda \int_{t-\tau}^t H(t, s) \cdot (x_M(s) - x(s)) ds,$$

and

$$R_{M_2}(t) = x_M(t) - x(t).$$

Remark 1 For any $t \in [-\tau, T]$, if $R_{M_1}(t) = 0$ and $R_{M_2}(t) = 0$, then $x(t) = x_M(t)$; if $\lim_{M \rightarrow \infty} R_{M_1}(t) = 0$ and $\lim_{M \rightarrow \infty} R_{M_2}(t) = 0$, then $\lim_{M \rightarrow \infty} x_M(t) = x(t)$.

Remark 2 For any $t \in [-\tau, T]$, if $R_{M_1}(t) \equiv 0$ and $R_{M_2}(t) \equiv 0$, then $x_M(t)$ is an exact solution of Eqs. (4) and (5); if $\lim_{M \rightarrow \infty} R_{M_1}(t) = 0$ and $\lim_{M \rightarrow \infty} R_{M_2}(t) = 0$, then $x_M(t)$ converges to the exact solution of Eqs. (4) and (5).

Remark 3 If $\lim_{M \rightarrow \infty} \frac{1}{2} \int_{-\tau}^0 T_2^2(t, x_M(t)) dt + \frac{1}{2} \int_0^T T_1^2(t, x_M(t)) dt = 0$, then the approximation solution $x_M(t)$ converges to the exact solution $x(t)$ of Eqs. (4) and (5).

To compute the integrals from Eqs. (4) and (5), we consider the uniform partition of $[-\tau, T]$:

$$\Delta : -\tau = t_0 < t_1 < \dots < t_n = 0 < t_{n+1} < \dots < t_q = T$$

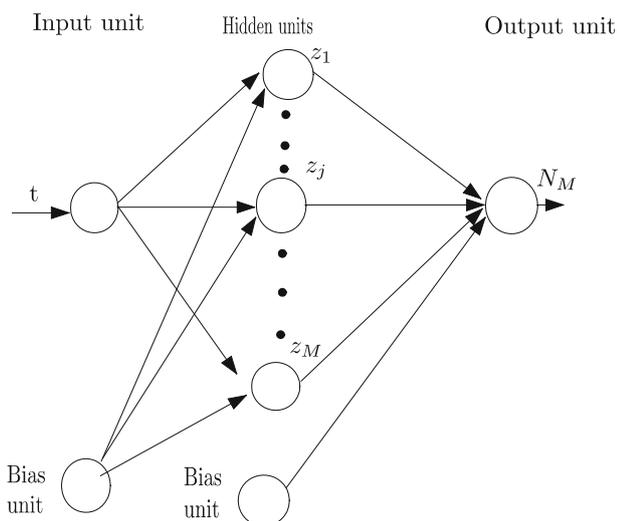


Fig. 2 NNM with one input unit and one output unit

with $q = (p + 1)n, t_i = t_{i-1} + \frac{\tau}{n} = -\tau + \frac{i\tau}{n}, i = \overline{1, q}$. On these knots, the terms of the sequence of successive approximations are the following:

$$\begin{aligned}
 x(t_i) &= \psi(t_i), \quad i = \overline{0, n}, \\
 x(t_i) &= g(t_i) + \lambda \int_{t_i-\tau}^{t_i} H(t_i, s) \cdot x(s) ds, \quad i = \overline{n + 1, q}.
 \end{aligned}
 \tag{11}$$

The numerical calculation can be implemented to determine the integration of Eq. (11). Let us consider a three-layered NNM (see Fig. 2) with one unit entry t , one hidden layer consisting of M activation functions, and one unit output $N_M(t, P)$. In this paper, we use the unipolar sigmoidal activation function $f(\cdot)$. Here, the dimension of NNM is $1 \times M \times 1$.

Hence, in our proposed UAM, the solution of HVDIEs can be simply obtained with the algorithm of optimization; in addition, the adjustable parameters of NNM are systematically updated in such a way. Hence, the problem formulation can be expressed as the typical minimization problem:

$$\text{Minimize } J(P) \tag{12}$$

where P is the vector containing all the adjustable parameters. For instance, we can use the penalty method for the minimization problem. Therefore, for solving HVDIEs by Eq. (12), we have:

$$\text{Minimize } \sum_{g=1}^G \frac{1}{2} T_2^2(t_g, x_M(t_g)) + \frac{1}{2} T_1^2(t_g, x_M(t_g)), \tag{13}$$

where G is the total number of points chosen within the domain of $[-\tau, T]$ and g is the point index.

Below, we present the following algorithm that gives the approximate solution using NNM:

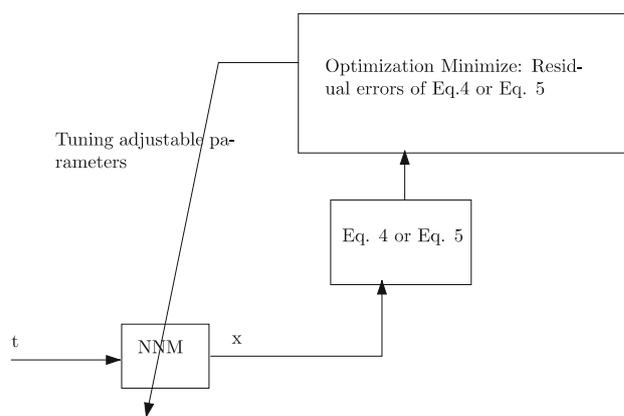


Fig. 3 Diagram of proposed NNM

Step 1 Choose the numbers of neurons and hidden layers for NNM as small as possible at the beginning.

Step 2 Apply an optimization technique to determine the sub-optimal adjustable parameters of NNM in such a way that the residual errors are minimized.

Step 3 If the residual errors are less than tolerance then stop.

If not, then try to increase the various numbers in Step 1 and go to Step 2.

Figure 3 shows the overall diagram of the proposed UAM in determining the solution of HVDIEs.

Numerical examples

In this section, two examples are given to illustrate the technique proposed in this paper.

Example 4.1 Consider the initial value problem:

$$x(t) = \begin{cases} e^{t-\tau} + \int_{t-\tau}^t x(s) ds, & t \in [0, 0.5], \\ \psi(t), & t \in [-\tau, 0], \end{cases}$$

with $\tau = 0.5, T = 0.5$, and $\psi: [-0.5, 0] \rightarrow \mathbb{R}$ is defined by $\psi(t) = e^t, t \in [-\tau, 0]$. The exact solution is $x: [-0.5, 0.5] \rightarrow \mathbb{R}$ given by $x(t) = e^t, t \in [-0.5, 0.5]$. Applying for $M = 9$, the above presented method, we obtain approximate solution with the error $3.21e^{-13}$. The exact and obtained solutions of Hammerstein–Volterra delay integral equation in this example are shown in Fig. 4. We see that the approximate solution obtained by the NNM has good accuracy on the whole interval. In Table 1, we compare the error of the present method (Method 1), method 2 in [13], method 3 in [5], method 4 in [52], and method 5 in [39].

Fig. 4 Compares the exact solution and obtained solution

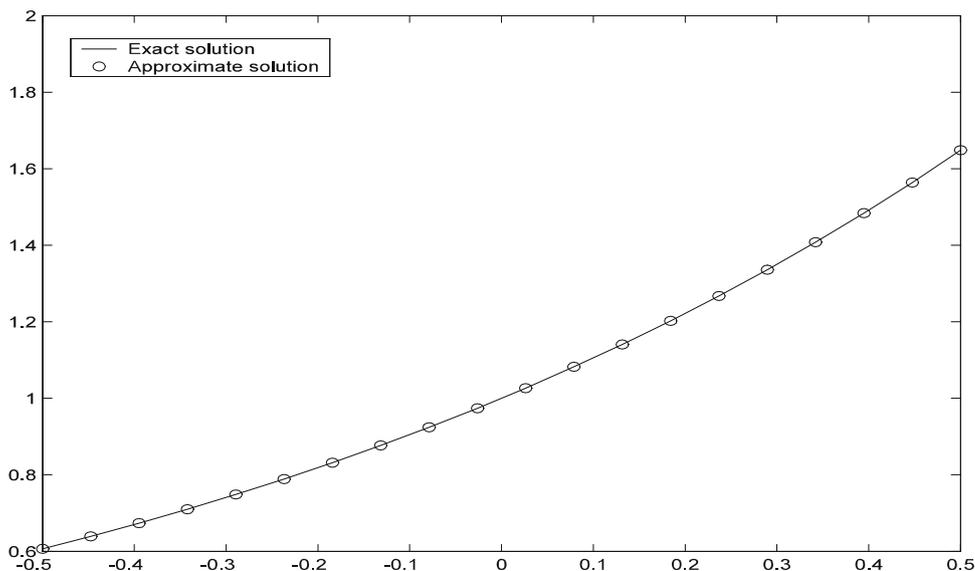


Table 1 Comparison of the errors for Example 4.1

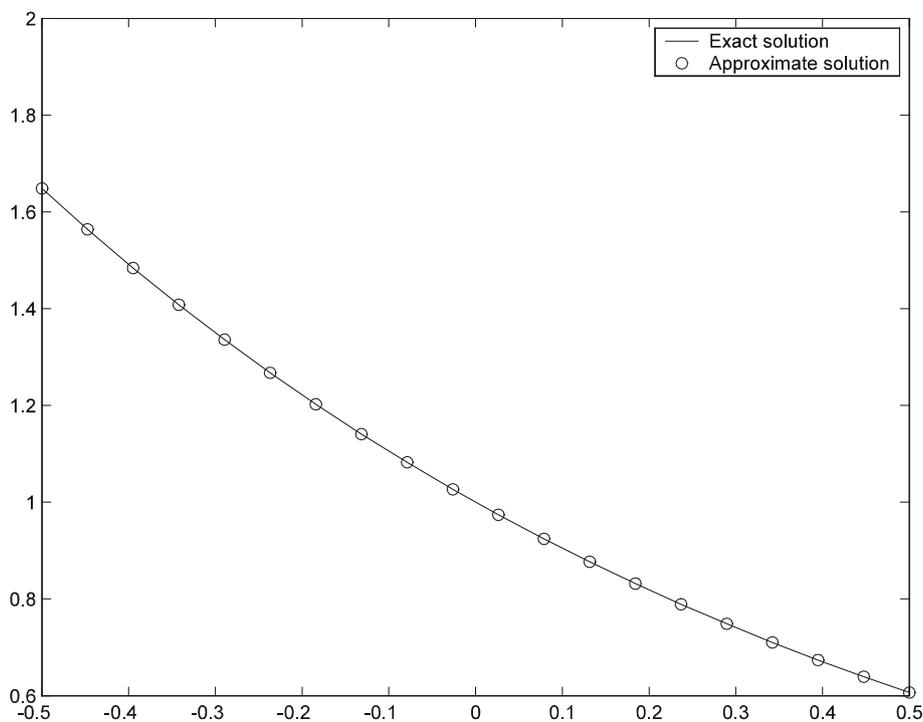
| M | Method 1 | Method 2 | Method 3 | Method 4 | Method 5 |
|-----|---------------|--------------|--------------|--------------|---------------|
| 8 | $2.22e^{-11}$ | $3.54e^{-6}$ | $4.11e^{-8}$ | $5.31e^{-6}$ | $2.01e^{-9}$ |
| 9 | $3.21e^{-13}$ | $6.21e^{-7}$ | $3.30e^{-8}$ | $3.01e^{-6}$ | $4.71e^{-10}$ |

new infected cases on unit time, and $g(t)$ be the proportion of immigrants that still have the disease at the moment t . Considering $P(s)$ as the probability of having the infection for a time at least s after infection, the spread of infection is governed by the integral equation:

Example 4.2 According to the epidemic model presented in [18], we consider that $x(t)$ be the proportion of infectious individuals at the moment s , $f(x(s))$ be the proportion of

$$x(t) = \begin{cases} g(t) + \int_{t-\tau}^t P(t-s) \cdot f(x(s)) ds, & t \in [0, T], \\ \psi(t), & t \in [-\tau, 0]. \end{cases}$$

Fig. 5 Compares the exact solution and obtained solution



Suppose that the proportion of new infected cases on unit time, $f(x(s))$ is proportional with $x(s)$, and P is a positive decreasing function with $P(0) = 1$. Let $\tau = T = 0.5$. Since it is natural to suppose that the proportion of immigrants that still have the disease is decreasing in time according to the decisions of the authorities, we were shown to the following model:

$$x(t) = \begin{cases} g(t) + \int_{t-\tau}^t e^{-(t-s)} \cdot x(s) ds, & t \in [0, 0.5], \\ \psi(t), & t \in [-0.5, 0], \end{cases}$$

with $g(t) = 0.5e^{-t}$ and $\psi(t) = e^{-t}$. The exact solution is $x: [-0.5, 0.5] \rightarrow \mathbb{R}$ given by $x(t) = e^{-t}$, $t \in [-0.5, 0.5]$. Applying, for $M = 10$, the above presented method, we obtain approximate solution with the error $4.23e^{-9}$. The exact and obtained solutions of Hammerstein–Volterra delay integral equation in this example are shown in Fig. 5.

Conclusions

Solving Hammerstein–Volterra delay integral equations using the universal approximators, that is, NNM was presented in this paper. We proposed NNM approximation method based on unipolar sigmoidal functions. The reliability and efficiency of the proposed method are demonstrated on the numerical experiments. In addition, we can execute this method in a computer simply.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Abbasbandy, S., Otadi, M.: Numerical solution of fuzzy polynomials by fuzzy neural network. *Appl. Math. Comput.* **181**, 1084–1089 (2006)
2. Abbasbandy, S., Otadi, M., Mosleh, M.: Numerical solution of a system of fuzzy polynomials by fuzzy neural network. *Inform. Sci.* **178**, 1948–1960 (2008)
3. Abbasbandy, S.: Application of He's homotopy perturbation method to functional integral equations. *Chaos, Solitons Fractals* **31**, 1243–1247 (2007)
4. Abbasbandy, S.: Numerical solutions of the integral equations: homotopy perturbation method and Adomian's decomposition method. *Appl. Math. Comput.* **173**, 493–500 (2006)
5. Avaji, M., Hafshejani, J.S., Dehcheshmeh, S.S., Ghahfarokhi, D.F.: Solution of delay Volterra integral equations using the variational iteration method. *J. Appl. Sci.* **12**, 196–200 (2012)
6. Babolian, E., Abbasbandy, S., Fattahzadeh, F.: A numerical method for solving a class of functional and two dimensional integral equations. *Appl. Math. Comput.* **198**, 35–43 (2008)
7. Baker, C.T.H.: A perspective on the numerical treatment of Volterra equations. *J. Comput. Appl. Math.* **125**, 217–249 (2000)
8. Belytschko, T., Krongauz, Y., Organ, D.: Meshless methods: an overview and recent developments. *Comput. Methods Appl. Mech. Eng.* **139**, 3–47 (1996)
9. Berenguer, M.I., Gamez, D., Garralda-Guillem, A.I., Galan, M.R., Serrano Perez, M.C.: Biorthogonal systems for solving Volterra integral equation systems of the second kind. *J. Comput. Appl. Math.* **235**, 1875–1883 (2011)
10. Bhrawy, A.H., Abdelkawy, M.A., Machado, J.T., Amin, A.Z.M.: Legendre–Gauss–Lobatto collocation method for solving multi-dimensional Fredholm integral equations. *Comput. Math. Appl.* (2016) (in press)
11. Bhrawy, A.H., Tohidi, E., Soleymani, F.: A new Bernoulli matrix method for solving high-order linear and nonlinear Fredholm integro-differential equations with piecewise intervals. *Appl. Math. Comput.* **219**, 482–497 (2012)
12. Biazar, J., Eslami, M., Islam, M.R.: Differential transform method for special systems of integral equations. *J. King Saud Univ. Sci.* **24**, 211–214 (2012)
13. Bica, A., Iancu, C.: A numerical method in terms of the third derivative for a delay integral equation from biomathematics. *J. Inequal. Pure Appl. Math.* **6**(42), 1–8 (2005)
14. Borzabadi, A.H., Fard, O.S.: A numerical scheme for a class of nonlinear Fredholm integral equations of the second kind. *J. Comput. Appl. Math.* **232**, 449–454 (2009)
15. Buckley, J.J., Hayashi, Y.: Can fuzzy neural nets approximate continuous fuzzy functions? *Fuzzy Sets Syst.* **61**, 43–51 (1994)
16. Chen, Y., Tang, T.: Spectral methods for weakly singular Volterra integral equations with smooth solutions. *J. Comput. Appl. Math.* **233**, 938–950 (2009)
17. Cheng, Y., Wang, J., Li, R.: The complex variable element-free Galerkin method for two-dimensional elastodynamics problems. *Int. J. Appl. Mech.* **4**, 1250042–1–125004223 (2012)
18. Cooke, K.L.: An epidemic equation with immigration. *Math. Biosci.* **29**, 135–158 (1979)
19. Hagan, M.T., Demuth, H.B., Beale, M.: Neural network design. PWS publishing company, Massachusetts (1996)
20. Hashemizadeh, E., Rostami, M.: Numerical solution of Hammerstein integral equations of mixed type using the Sinc-collocation method. *J. Comput. Appl. Math.* **279**, 31–39 (2015)
21. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, New Jersey (1999)
22. Hornick, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989)
23. Ishibuchi, H., Kwon, K., Tanaka, H.: A learning algorithm of fuzzy neural networks with triangular fuzzy weights. *Fuzzy Sets Syst.* **71**, 277–293 (1995)
24. Ito, Y., Saito, K.: Superposition of linearly independent functions and finite mappings by neural networks. *Math. Sci.* **21**, 27–33 (1996)
25. Kauthen, J.P.: Continuous time collocation method for Volterra–Fredholm integral equations. *Numer. Math.* **56**, 409–424 (1989)
26. Khanna, T.: Foundations of neural networks. Addison-Wesley, Reading, MA (1990)
27. Lapedes, A., Farber, R.: How neural nets work. In: Anderson, D.Z. (ed.) Neural information processing systems, pp. 442–456. American Institute of Physics, New York (1988)
28. Leephakpreeda, T.: Novel determination of differential-equation solution: universal approximation method. *J. Comput. Appl. Math.* **146**, 443–457 (2002)
29. Liew, K.M., Cheng, Y., Kitipornchai, S.: Boundary element-free method (BEFM) and its application to two-dimensional elasticity problems. *Int. J. Numer. Methods Eng.* **65**, 1310–1332 (2006)



30. Linz, P.: Analytical and numerical methods for Volterra equations. SIAM, Philadelphia, PA (1985)
31. Lippmann, R.P.: An introduction to computing with neural nets. IEEE ASSP Mag. **4**, 4–22 (1987)
32. Malek, A., Beidokhti, R.S.: Numerical solution for high order differential equations using a hybrid neural network-Optimization method. Appl. Math. Comput. **183**, 260–271 (2006)
33. Maleknejad, K., Basirat, B., Hashemizadeh, E.: Hybrid Legendre polynomials and Block-Pulse functions approach for nonlinear Volterra–Fredholm integro-differential equations. Comput. Math Appl. **61**, 2821–2828 (2011)
34. Mosleh, M.: Fuzzy neural network for solving a system of fuzzy differential equations. Appl. Soft Comput. **13**, 3597–3607 (2013)
35. Mosleh, M.: Numerical solution of fuzzy linear Fredholm integro-differential equation by fuzzy neural network. Iran. J. Fuzzy Syst. **11**, 91–112 (2014)
36. Mosleh, M., Otadi, M.: Simulation and evaluation of fuzzy differential equations by fuzzy neural network. Appl. Soft Comput. **12**, 2817–2827 (2012)
37. Mosleh, M., Otadi, M., Abbasbandy, S.: Fuzzy polynomial regression with fuzzy neural networks. Appl. Math. Model. **35**, 5400–5412 (2011)
38. Mosleh, M., Allahviranloo, T., Otadi, M.: Evaluation of fully fuzzy regression models by fuzzy neural network. Neural Comput Appl. **21**, 105–112 (2012)
39. Mosleh, M., Otadi, M.: Least squares approximation method for the solution of Hammerstein–Volterra delay integral equations. Appl. Math. Comput. **258**, 105–110 (2015)
40. Mosleh, M., Otadi, M., Abbasbandy, S.: Evaluation of fuzzy regression models by fuzzy neural network. J. Comput. Appl. Math. **234**, 825–834 (2010)
41. Otadi, M.: Fully fuzzy polynomial regression with fuzzy neural networks. Neurocomputing **142**, 486–493 (2014)
42. Otadi, M., Mosleh, M.: Simulation and evaluation of dual fully fuzzy linear systems by fuzzy neural network. Appl. Math. Model. **35**, 5026–5039 (2011)
43. Otadi, M., Mosleh, M., Abbasbandy, S.: Numerical solution of fully fuzzy linear systems by fuzzy neural network. Soft. Comput. **15**, 1513–1522 (2011)
44. Otadi, M., Mosleh, M.: Simulation and evaluation of interval-valued fuzzy linear Fredholm integral equations with interval-valued fuzzy neural network. Neurocomputing **205**, 519–528 (2016)
45. Otadi, M., Mosleh, M.: Numerical solution of quadratic Riccati differential equation by neural network. Math. Sci. **5**, 249–257 (2011)
46. Picton, P.: Neural networks, 2nd edn. Palgrave, Great Britain (2000)
47. Ren, H., Cheng, Y.: The interpolating element-free Galerkin (IEFG) method for two-dimensional potential problems. Eng. Anal. Boundary Elem. **36**, 873–880 (2012)
48. Schalkoff, R.J.: Artificial Neural Networks. McGraw-Hill, New York (1997)
49. Sorkun, H.H., Yalcinbas, S.: Approximate solutions of linear Volterra integral equation systems with variable coefficients. Appl. Math. Model. **34**, 3451–3464 (2010)
50. Stanley, J.: Introduction to Neural Networks, 3rd edn. Sierra Madre (1990)
51. Wang, Q., Wang, K., Chen, S.H.: Least squares approximation method for the solution of Volterra–Fredholm integral equations. J. Comput. Appl. Math. **272**, 141–147 (2014)
52. Yalcinbas, S.: Taylor polynomial solutions of nonlinear Volterra–Fredholm integral equations. Appl. Math. Comput. **127**, 195–206 (2002)
53. Zheng, B., Dai, B.: A meshless local moving Kriging method for two-dimensional solids. Appl. Math. Comput. **218**, 563–573 (2011). [**199–249 (1975)**]

