Int. J. Nonlinear Anal. Appl. 15 (2024) 10, 235–241
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2023.31609.4681

A new approach for solving delay differential equations with time varying delay

Tahereh Jabbari Khanbehbin[a], Sohrab Effati[a,b,*], Morteza Gachpazan[a], Seyed Mohsen Miri[a]

[a]*Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran*

[b]*Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP), Ferdowsi University of Mashhad, Mashhad, Iran*

(Communicated by Saeid Abbasbandy)

Abstract

In this paper, Artificial Neural Networks are used to solve Delay Differential Equations. We have suggested an appropriate approximation function based on ANN and then by solving an optimization problem of error function, the neural network is trained. The advantage of this technique is that the proposed approximation functions, with a slight modification, can be used for most types of delay differential equations, including DDE with constant delay, time dependent delay and pantograph delay. To demonstrate the effectiveness of the method, various examples have been tested and validity and efficiency of the method have been shown.

Keywords: Delay Differential Equations, Artificial Neural Networks, unconstrained optimization problem
2020 MSC: 34K06, 34K28, 68T05

1 Introduction

The purpose of this paper is to provide a proper method for solving Delay Differential Equations(DDEs). The history of the DDEs goes back about 200 years ago. After Picard's lecture on the impact of delay on physical device modeling at the International Mathematics Conference in 1908, attention was drawn to this issue. Of course, engineers were aware of the importance and impact of the delay on physical models, but due to inadequate theoretical discussions, they discarded its application. until Krasovsky and Bellman and others published books that provided a clearer picture of this theory. There are many books in the field of DDEs and their applications such as: [5, 7, 13, 14, 16, 23].

Since many natural phenomena are modeled with DDEs, researchers were interested in solving these equations. In [3] you can find several applications and methods for solving DDEs until 1995. In 2000 Shampine in [28] showed how to solve DDEs with dde23 software. After that several numerical methods were suggested for these equations, including the Runge Kutta methods [5], the spline methods [11, 26], the variational iteration method [22, 24, 29], Adomian decomposition method [12], the homotopy methods [27]. Also a class of methods for solving DDEs is the use of polynomial series and orthogonal functions, this technique simplifies these problems by reducing them to solving a system of algebraic equations, thereby greatly simplifying them. Some of these methods are: hybrid of block-pulse functions and Bernstein polynomials [4, 21], Haar wavelet [2], the Galerkin method [17], Hermit wavelet [19] and etc.

*Corresponding author
Email addresses: jabbari.tahere@gmail.com (Tahereh Jabbari Khanbehbin), s-effati@um.ac.ir (Sohrab Effati), gachpazan@um.ac.ir (Morteza Gachpazan), mohsenmiri80@gmail.com (Seyed Mohsen Miri)

Received: August 2023 *Accepted:* November 2023

We were looking for a method that it can solve DDEs with time dependent delay. For this aim we implement an Artificial Neural Network (ANN). In 1944, two researchers from Chicago University named McCullough and Walter Pitts presented the first model of neural networks, they were the ones who founded the first cognitive science department at MIT in 1952. The perceptron was the first trainable neural network proposed by Cornell University psychologist Frank Rosenblatt in 1957. Its design was very similar to modern neural networks, but the Rosenblatt perceptron had only one layer with adjustable weights placed between the input and output layers.

Recently, ANNs are considerable as a effective tools for function approximation [1, 18]. Mathematicians proved that continuous functions can be approximated by a multi-layer perceptron on the basis of a compact set of R^n. In a theorem in [15], Gybenko proves that a Neural Network approximation with a sigmoid active function can approximate continuous functions successfully. For the first time, in order to solve PDEs and ODEs, Lagaris et al. proposed using neural networks [20]. In their study, they used a multilayer perceptron neural network. Effati and Pakdaman in [9] used ANNs for approximating the fuzzy differential equations' solution. Furthermore, they approximate state, co-state, and control functions for optimal control problems in [10]. Sabouri et al. in [25] can solve fractional optimal control problems with Neural Networks and etc. In another work Effati et al. [8] implement an ANN to solve a linear optimal control problem with quadratic cost functional and fuzzy variables. Recently, Bhagya and Dash in [6] discussed a variety of applications of ANN to the modeling of nonlinear problems in food engineering.

As far as we know, there is no prior work on solving DDEs with time dependent delays by artificial neural networks. Therefore, we attempt to implement the ability of neural networks to approximate these equations.

This paper is organized as follow:

Introduction of DDEs and different types of these equations are discussed in Section 2. In Section 3 we design a new Neural Network to solve delay differential equations. Section 4 presents three types of examples to illustrate the effectiveness of the proposed method and the paper ends in Section 5 with the conclusion.

2 Delay Differential Equations

A wide class of such model is represented by the following general form of DDEs,

$$\begin{cases} \dot{x}(t) = f(t, x(t), x(\alpha(t))), & t_0 \le t \le t_f \\ x(t) = \phi(t), & \alpha(t_0) \le t \le t_0, \end{cases} \tag{2.1}$$

where $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, $\alpha(t) \leqslant t$ and $\phi(t)$ represents the initial information of the variable x where $\phi \in C^0[\alpha(t_0), t_0]$. If $\alpha(t) = t - \tau$ where τ is constant and $\tau > 0$, the above equation is called a DDE with constant delay. In the case of $\alpha(t) = t - \tau(t)$ where $\tau(t) \ge 0$ is a given function with $0 < \dot{\tau}(t) < 1$, we have a DDE with time depenedent delay and if $\alpha(t) = kt$ with $k \in (0, 1)$, equation (2.1) is called a pantograph delay equation. In this case $kt = t - \tau(t)$ with $\tau(t) = (1 - k)t$. For notational convenience we use a single delay, the generalization to multiple delays is straightforward.

Existence and uniqueness theorems for DDE (2.1), as with ordinary differential equations, are essentially on the basis of the continuity of the function $f(t, u, v)$ with respect to t and Lipschitz continuity with respect to u, v.

Theorem 2.1. [5, Theorem 2.2.1](Local existence) Consider the equation

$$\begin{cases} \dot{x}(t) = f(t, x(t), x(t - \tau(t))), & t_0 \le t \le t_f \\ x(t) = \phi(t), & t_0 - \tau(t_0) \le t \le t_0, \end{cases} \tag{2.2}$$

and assume that the function $f(t, u, v)$ is continuous on $A \subseteq [t_0, t_f) \times \mathbb{R}^n \times \mathbb{R}^n$ and locally Lipschitz continuous with respect to u and v. Moreover, assume that the delay function $\tau(t) \geqslant 0$ is continuous in $[t_0, t_f)$, $\tau(t_0) = 0$ and, for some $\xi > 0$, $t - \tau(t) > t_0$ in the interval $(t_0, t_0 + \xi]$. Then there is a unique solution to the problem (2.2) in $[t_0, t_0 + \delta)$ for some $\delta > 0$ and this solution is continuously dependent on the initial data.

Under the same hypotheses, it can be seen that it is possible to continue until a maximum solution is defined in the interval $[t_0, b)$, with $t_0 < b \leqslant t_f$. The following theorem about global existence can be derived from this.

Theorem 2.2. [5, Theorem 2.2.2](Global existence) Under the hypotheses of Theorem 2.1, there exists a unique maximal solution for (2.2) on the entire interval $[t_0, t_f)$, if it is bounded.

Before we can apply the global existence theorem, we need to have a priori bound for the solution. A corollary to this can be found bellow.

Corollary 2.3. [5, Corollary 2.2.3] In addition to the hypotheses of Theorem 2.1, suppose that the function $f(t,u,v)$ satisfies the condition

$$||f(t,u,v)|| \leqslant M(t) + N(t)(||u|| + ||v||),$$

in $[t_0, t_f) \times \mathbb{R}^n \times \mathbb{R}^n$, where $M(t)$ and $N(t)$ are continuous positive functions on $[t_0, t_f)$. Hence, there is only one solution to (2.2) on the entire interval $[t_0, t_f)$ and it is unique.

3 Design of a Neural Network

In this section, we propose a two layer neural network model to solve Delay Differential Equations. The mathematical structure of this neural network can be considered as follows.

$$N(t, W) = \sum_{i=1}^{k} \nu_i \sigma(\theta_i), \quad \theta_i = w_i t + b_i, \tag{3.1}$$

where ν^i for $i = 1, ...k$, are the weights of output layer, b^i are bias weights and w^i are the weights of input layer where $W = (v^i, w^i, b^i)$ that contains all neural network weights. σ is considered as an arbitrary activation function, here we implemented the sigmoid function in the numerical examples, as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{3.2}$$

Gybenko in [15] showed that the finite sum (3.1) is a universal approximator.

Theorem 3.1. Assume that $\sigma(.)$ is a continuous sigmoidal function. It follows that finite sum of the form:

$$M(t) = \sum_{i=1}^{k} \nu_i \sigma(\theta_i),$$

is dense over an interval I, in the space of continuous functions (denoted by $C(I)$)(where $\theta_i = w_i^T t + b_i$). To put it another way, for any $\epsilon > 0$ and continuous function $m(.)$, there is a sum $M(.)$, of the mentioned form which:

$$|M(t) - m(t)| < \epsilon, \quad \text{for all } t \in I. \tag{3.3}$$

We try to propose appropriate approximate function for $x(t)$ to solve the equation (2.1). Our suggested approximation is as follows:

$$x_N(t, W) = \begin{cases} \phi(t) + \psi(t) N(t, W), & t \geq t_0 \\ \phi(t), & t < t_0 \end{cases} \tag{3.4}$$

If $\psi, \phi \in C^1[t_0, t_f]$ then

$$\dot{x}_N(t, W) = \begin{cases} \dot{\phi}(t) + \psi(t) \dot{N}(t, W) + \dot{\psi}(t) N(t, W), & t \geq t_0 \\ \dot{\phi}(t), & t < t_0 \end{cases} \tag{3.5}$$

If $\psi(t_0) = \dot{\psi}(t_0) = 0$ then $\dot{x}_N(t, W)$ is a continues function, so $x_N(t, W) \in C^1[t_0, t_f]$. With proposing this approximation function for $x(t)$ and substituting it in DDE (2.1), we have

$$\dot{x}_N(t, W) = f(t, x_N(t, W), x_N(\alpha(t), W)), \quad t \geq t_0 \tag{3.6}$$

For solving (3.6), it is introduced the squared residual error function in the following:

$$E_W(t) = \dot{x}_N(t, W) - f(t, x_N(t, W), x_N(\alpha(t), W)), \quad t_0 \leq t \leq t_f$$

As an approximation of the neural network error, we have:

$$R(W, X) = \sum_{t_k \in X} E_W^2(t_k), \tag{3.7}$$

where $X = \{x_i \in [t_0, t_f],\ x_{i+1} = x_0 + ih,\ 0 \leqslant i \leqslant n-1\}$. Then we must solve the following unconstrained optimization problem for minimizing neural weights:

$$\min_{W} R(W, X) = \sum_{t_k \in X} E_W^2(t_k), \tag{3.8}$$

Any classical mathematical optimization algorithm and heuristic approach can be used to solve problem (3.8) that is an unconstrained optimization problem. We have used of matlab optimization packages. The proposed method offer some of advantages, including:

- The most important advantage is that the solution of DDEs is presented as a continuous and differentiable function that satisfies the primary or final conditions. In addition, the approximated solution can be computed point-wise at any point within the training interval, even among the training points.

- The precision of the solutions are acquired by an increase in the number of neurons or the use of more accurate training or optimization algorithms.

- Theorem 3.1 is the basis of the suggested approximation method. It is inevitable that the weights obtained from neural networks are convergent to optimal values,because of the neural networks are worldwide approximators. An illustration of this concept can be seen in numerical examples.

- The proposed method can solve most types of DDEs with acceptable solutions, including: DDEs with constant delay or time varying delay and linear or nonlinear DDEs with same algorithm.

4 Numerical Examples

In this section, we present numerical examples and in order to show the efficiency of the proposed method, we have reported three types of errors for each example. The first is $\tilde{R}$, the approximated value of R in unconstrained optimization problem (3.8), the second reported error is the maximum error of x i.e the maximum absolute value of the difference between the approximate value of x, calculated from the proposed neural network method and its exact value, in the training points, with given $L = \frac{t_f - t_0}{h}$,

$$\delta = \max_{0 \leq i \leq L} |x(t_i) - x^*(t_i)|,$$

and the third error is the global error, which is the error of x throughout the interval $[t_0, t_f]$,

$$\Delta = \int_{t_0}^{t_f} |x(t) - x^*(t)| dt.$$

Example 4.1. Take into consideration the following delay differntial equation with constant delay:

$$\begin{cases} \dot{x} = 1 + x(t - \frac{\sqrt{2}}{2}), & 0 \leq t \leq 1 \\ x(t) = 0, & -\frac{\sqrt{2}}{2} \leq t < 0 \\ x(0) = 1. & \end{cases} \tag{4.1}$$

the analytical solution of (4.1) is:

$$x^*(t) = \begin{cases} 1 + t, & 0 \leq t \leq \frac{\sqrt{2}}{2} \\ \frac{5}{4} + \frac{\sqrt{2}}{2} - \sqrt{2} + (2 - \frac{\sqrt{2}}{2})t + \frac{t^2}{2}, & \frac{\sqrt{2}}{2} \leq t \leq \sqrt{2} \end{cases} \tag{4.2}$$

In (3.4) we choose $\psi(t) = t$. with proposing the following approximation function for $x(t)$,

$$x_N(t) = \begin{cases} 1 + tN(t), & t \geq 0 \\ 0, & t < 0. \end{cases} \tag{4.3}$$

In this case, the neural network is trained to solve problem in the interval $[0, \sqrt{2}]$ with $k = 10$ neurons and 30 training points. In the validation points, the neural network solution has these errors: $\tilde{R} = 1.8 \times 10^{-3}$, local error $\delta = 3.9 \times 10^{-3}$, global error $\Delta = 7.7 \times 10^{-4}$. Also, we depict the approximate and exact values of $x(t)$ for this example in Fig. 1.

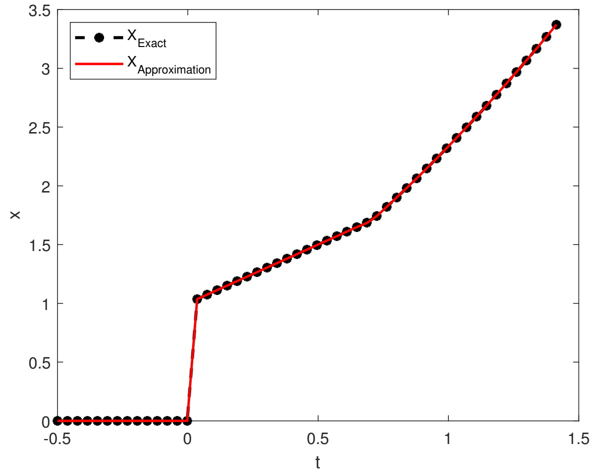

Figure 1: Exact and approximated solution of $x(t)$ for Example 4.1

Example 4.2. Consider the following pantograph delay differntial equation

$$\begin{cases} \dot{x} = \frac{1}{2}(e^{\frac{t}{2}}x(\frac{t}{2}) + x(t)), & t \geq 0 \\ x(t) = 1, & t \leq 0. \end{cases} \tag{4.4}$$

According to [12], the exact solution of equation (4.4) is $x^*(t) = e^x$. With selecting $\psi(t) = t^2$ and proposing the following approximation function for $x(t)$,

$$x_N(t) = \begin{cases} 1 + t^2 N(t), & t \geq 0 \\ 1, & t \leq 0. \end{cases} \tag{4.5}$$

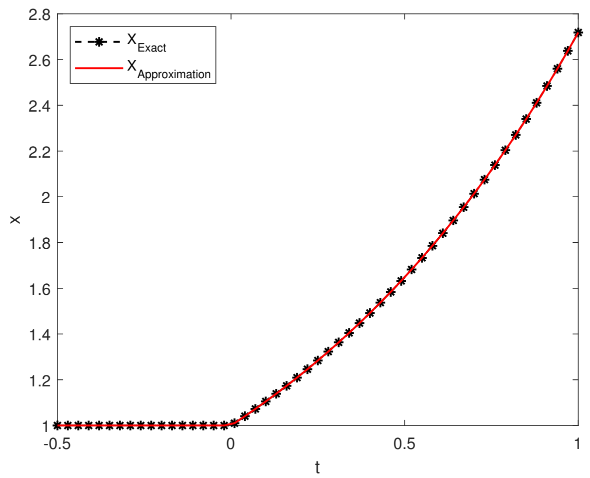

Figure 2: Exact and approximated solution of $x(t)$ for Example 4.2

The neural network is trained to solve in the interval $[0, 1]$ with 10 neurons and 30 training points and in the validation points, the NN solution errors are: $\tilde{R} = 5.4 \times 10^{-6}$, local error $\delta = 5.2 \times 10^{-5}$ and global error is $\Delta = 2.1 \times 10^{-5}$. Fig. 2 shows the approximate and exact solution of $x(t)$ for equation (4.4).

Example 4.3. Consider the following delay differntial equation with time dependent delay:

$$\begin{cases} \dot{x} = -x(t - \frac{1}{2} - \frac{e^{-t}}{2}), & t \geq 0 \\ x(t) = 1, & t \leq 0. \end{cases} \tag{4.6}$$

With proposing the following approximation function for $x(t)$:

$$x_N(t) = \begin{cases} 1 + tN(t), & t \geq 0 \\ 1, & t \leq 0 \end{cases} \tag{4.7}$$

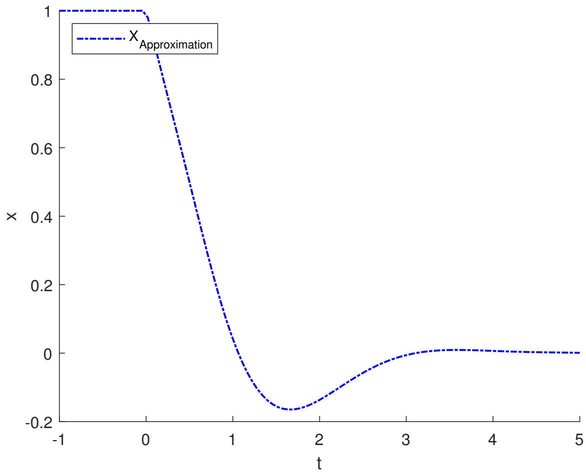

Figure 3: approximated solution of $x(t)$ for Example 4.3

For solving this problem in the interval $[0, 5]$, we train the neural network with 10 neurons and 20 training points. The error of neural network solution is $\tilde{R} = 1.8 \times 10^{-4}$. In this example, since there is no analytical result for $x(t)$, so we cannot report the local and global errors. The approximated solution of $x(t)$ is depicted in Fig. 3.

5 Conclusion

In this paper, we applied an approximation function based on neural network model to solve DDEs. The advantage of this approximated function is that it can be used for three types of DDEs with a slight modification. To calculate the neural network weights, we construct an unconstrained optimization problem and solve it with a mathematical optimization algorithm. Various numerical examples are presented to show the high efficiency and accuracy of the suggested method. Further research can be done on DDEs with state dependent delay or DDEs of neutral type.

References

[1] S. Abbasbandy, M. Otadi, and M. Mosleh, *Numerical solution of a system of fuzzy polynomials by fuzzy neural network*, Inf. Sci. **178** (2008), no. 8, 1948–1960.

[2] I. Aziz and R. Rohul Amin, *Numerical solution of a class of delay differential and delay partial differential equations via haar wavelet*, Appl. Math. Modell. **40** (2016), no. 23-24, 10286–10299.

[3] C. Baker, C. Paul, and D. Wille, *A bibliography on the numerical solution of delay differential equations*, Technical Report 269, University of Manchester, 1995.

[4] M. Behroozifar and S.A. Yousefi, *Numerical solution of delay differential equations via operational matrices of hybrid of block-pulse functions and Bernstein polynomials*, Comput. Meth. Differ. Equ. **1** (2013), no. 2, 78–95.

[5] A. Bellenand M. Zennaro, *Numerical Methods for Delay Differential Equations*, Oxford University Press, 2013.

[6] G. Bhagya Raj and K.K. Kshirod K Dash, *Comprehensive study on applications of artificial neural network in food process modeling*, Critic. Rev. Food Sci. Nutr. **62** (2022), no. 10, 2756–2783.

[7] R.D. Driver, *Ordinary and Delay Differential Equations*, volume 20. Springer Science & Business Media, 2012.

[8] S. Effati, M. Mansoori, and M. Eshaghnezhad, *Linear quadratic optimal control problem with fuzzy variables via neural network*, J. Exper. Theor. Artific. Intell. **33** (2021), no. 2, 283–296.

[9] S. Effati and M. Pakdaman, *Artificial neural network approach for solving fuzzy differential equations*, Inf. Sci. **180** (2010), no. 8, 1434–1457.

[10] S. Effati and M. Pakdaman, *Optimal control problem via neural networks*, Neural Comput. Appl. **23** (2013), no. 7-8, 2093–2100.

[11] A. El-Safty, M.S. Salim, and M.A. El-Khatib, *Convergence of the spline function for delay dynamic system*, Int. J. Comput. Math. **80** (2003), no. 4, 509–518.

[12] D.J. Evans and K.R. Raslan, *The Adomian decomposition method for solving delay differential equation*, Int. J. Comput. Math. **82** (2005), no. 1, 49–54.

[13] E. Fridman, *Introduction to Time-Delay Systems: Analysis and Control*, Springer, 2014.

[14] K. Gopalsamy, *Stability and Oscillations in Delay Differential Equations of Population Dynamics*, vol. 74, Springer Science & Business Media, 2013.

[15] G. Gybenko, *Approximation by superposition of sigmoidal functions*, Math. Control Signals Syst. **2** (1989), no. 4, 303–314.

[16] A. Halanay, *Differential Equations: Stability, Oscillations, Time Lags*, vol. 6, Elsevier, 1966.

[17] C. Hwang and M.Y. Chen, *Analysis of time-delay systems using the Galerkin method*, Int. J. Control, **44** (1986), no. 3, 847–866.

[18] A. Jafarian, S. Measoomy, and S. Abbasbandy, *Artificial neural networks based modeling for solving Volterra integral equations system*, Appl. Soft Comput. **27** (2015), 391–398.

[19] A. Kheirabadi, A.M.V aziri, and S. Effati, *Numerical solution of time-delay systems by Hermite wavelet*, Int. J. Dyn. Syst. Differ. Equ. **11** (2021), no. 1, 1–17.

[20] I. Lagaris, A. Likas, and D. Fotiadis, *Artificial neural networks for solving ordinary and partial differential equations*, IEEE Trans. Neural Networks **9** (1998), no. 5, 987–1000.

[21] H.R. Marzban and M. Razzaghi, *Solution of time-varying delay systems by hybrid functions*, Math. Comput. Simul. **64** (2004), no. 6, 597–607.

[22] S.T. Mohyud-Din and A. Yildirim, *Variational iteration method for delay differential equations using he's polynomials*, Z. Naturfor. A **65** (2010), no. 12, 1045–1048.

[23] F. Rihan, *Delay Differential Equations and Applications to Biology*, Springer, 2021.

[24] A. Saadatmandi and M. Dehghan, *Variational iteration method for solving a generalized pantograph equation*, Comput. Math. Appl. **58** (2009), no. 11-12, 2190–2196.

[25] J. Sabouri, S. Effati, and M. Pakdaman, *A neural network approach for solving a class of fractional optimal control problems*, Neural Process. Lett. **45** (2017), no. 1, 59–74.

[26] M. Shadia, *Numerical solution of delay differential and neutral differential equations using spline methods*, Ph. D. Thesis, Assuit University, Asyut, Egypt, 1992.

[27] F. Shakeri and M. Dehghan, *Solution of delay differential equations via a homotopy perturbation method*, Math. Comput. Modell. **48** (2008), no. 3-4, 486–498.

[28] L.F. Shampine, S. Thompson, and J. Kierzenka, *Solving delay differential equations with dde23*, available at `http://www.runet.edu/~{}thompson/webddes/tutorial.pdf`.

[29] T.L. Yookesh, E.D. Boobalan, and T.P. Latchoumi, *Variational iteration method to deal with time delay differential equations under uncertainty conditions*, Int. Conf. Emerg. Smart Comput. Inf. (ESCI), IEEE, 2020, pp. 252–256.