Al-Nahrain Journal of Science
ANJS, Vol.25 (3), September, 2022, pp. 25-32

Artificial Neural Network Technique for Solving Variable Order Fractional Integro-Differential Algebraic Equations

Ahmed N. Talib* and Osama H. Mohammed

Department of Mathematics and Computer Applications, College of Science, Al-Nahrain University, Baghdad, Iraq

Article's Information

Received: 19.06.2022
Accepted: 12.08.2022
Published: 30.09.2022

Keywords:
Artificial neural network
Variable order fractional derivative
Integro-differential algebraic equation
Fractional calculus

Abstract

In this paper, we will use an artificial neural network (ANN) to solve the variable order fractional integro-differential algebraic equations (VFIDAEs), which is a three-layer feed-forward neural architecture that is formed and trained using a backpropagation unsupervised learning algorithm based on the gradient descent rule for minimizing the error function and parameter modification (weights and biases). When we combine the initial conditions with the ANN output, we get a good approximation of the VFIDAE solution. Finally, the analysis is complemented by two numerical examples that demonstrate the method capability. The collected results show that the suggested strategy is quite successful, resulting in superior approximations in these cases.

DOI: 10.22401/ANJS.25.3.05
Corresponding author: ahmednatiq89@gmail.com

1. Introduction

Fractional calculus (FC) was successfully implemented in science and engineering [1-6]. Clearly the exact solutions of the fractional ordinary, partial and integro-differential equations are difficult, therefore numerical or even approximate algorithm are needed. Various numerical techniques have been developed to solve these equations. These methods include finite difference, Adomian decomposition, variational iteration, Laplace transforms and operational matrix methods [7-17].

The topic of variable order calculus has recently been considered. Samko and Ross (1993) presented the variable order derivative in 1993. Several methods for dealing with numerical calculations of both variable order ordinary and integro-differential equations have been presented. [18].

Because of its superior learning capacity, different machine intelligence processes, particularly artificial neural network (ANN) approaches, have emerged as a potent methodology for solving a range of real-world issues in recent years [19,20].

The advantages of the ANN technique, such as learning, adaptiveness, error computation, and fault-tolerance, have received a lot of attention [21-23]. The research of ANN for solving ordinary and partial differential equations is now receiving a lot of interest [24-29].

The FIDAEs are a generalization of integral and integro-differential equations with algebraic constraints which arise in the problem of evaluation of a chemical reaction within a small cell [30], dynamic processes in chemical reactors [31], identification of memory kernels in heat conduction and viscoelasticity [32] and etc. [33-37].

The purpose of this paper is to find numerical solution of variable order fractional integro-differential algebraic equations (VFIDAEs) using ANN.

Although several authors have looked into analytical and numerical methods for solving integral and integer order integro-differential equations, developing appropriate methods for solving variable fractional order integro-differential algebraic equations is, as far as we know, a new topic in the literature. We use an ANN approach to solve VFIDAEs since analytical techniques cannot always provide the precise solution.

2. Preliminaries

In this section, we recall some definitions and general concepts related to fractional calculus which may be used further in this paper [38-41].

Definition 2.1 (Conformable variable-order fractional derivative). The (left) conformable variable-order fractional derivative from a of a function f: $[a, \infty) \to$ R of order α: $[a, \infty) \to (0, 1]$ is defined by:

$$CD_a^{\alpha(t)} f(t) = \lim_{\varepsilon\to 0} \frac{f(t+\varepsilon(t-a)^{1-\alpha(t)})-f(t)}{\varepsilon}, t > a$$

when $a = 0$, ones write $CD^{\alpha(t)}$.

Let α: $[a,\infty) \rightarrow (0, 1]$ and $CD_a^{\alpha(t)}$ denotes the conformable fractional derivative. Then $\forall\ t > a$, the following properties are holds

1. $CD_a^{\alpha(t)}(kf + dg) = kCD_a^{\alpha(t)}(f) + dCD_a^{\alpha(t)}(\mathrm{g})\ \forall\ k, d\ \in \mathbb{R}$.
2. $CD_a^{\alpha(t)}(f\mathrm{g}) = fCD_a^{\alpha(t)}(\mathrm{g}) + \mathrm{g}CD_a^{\alpha(t)}(f)$.
3. $CD_a^{\alpha(t)}\left(\frac{f}{\mathrm{g}}\right) = \frac{\mathrm{g}CD_a^{\alpha(t)}(f) - fCD_a^{\alpha(t)}(\mathrm{g})}{\mathrm{g}^2}\ if\ \mathrm{g} \neq 0$.
4. If f is differentiable, then $CD_a^{\alpha(t)}(f) = (t-a)^{1-\alpha(t)}\frac{df}{dt}$.

3. The Approach

Let us consider the following (VFIDAEs)

$$CD_a^{\alpha_i(x)}u_i(x) = \sum_{i=1}^{n}\int_0^x k_i(x,t)\,u_i(t)dt + f_i(x),\quad \forall\ i = 1,2,\dots,n-1 \quad \dots(1)$$

such that:

$$g(x, u_1, u_2, \dots, u_n) = 0 \quad \dots(2)$$

With the initial condition:

$$u_i(x_0) = a_i\ , i = 1,2,\dots,n \quad \dots(3)$$

Let $u_{iN}(x,\Omega)$ represents the approximate solution of ANN model with Ω is a vector containing corresponding weights and x is the input data. Problem (1)-(3) is transformed into the following problem:

$$CD_a^{\alpha_i(x)}u_{iN}(x,\Omega) = \sum_{i=1}^{n}\int_0^x k_i(x,t)\,u_{iN}(t,\Omega)dt + f_i(x)\ ,\ \forall\ i = 1,2,\dots,n-1 \quad \dots(4)$$

such that:

$$g\big(x, u_{1N}(x,\Omega), u_{2N}(x,\Omega), \dots, u_{nN}(x,\Omega)\big) = 0 \quad \dots(5)$$

The approximate solution of ANN may be written as

$$u_{iN}(x,\Omega) = a_i + (x - x_0)N_i(x,\Omega), i = 1,2,\dots,n \dots(6)$$

The first term a_i in the right-hand side does not contain adaptable parameters and satisfies only initial/boundary conditions, whereas the second term $(x - x_0)N_i(x,\Omega)$ contain the single output $N_i(x,\Omega)$ of feed forward neural network with input x and vector containing the corresponding weights Ω .

Consider a three-layer ANN with one input node x, one hidden layer consisting of m number of nodes and one output node $N_i(x,\Omega)$.

The output $N_i(x,\Omega)$ is expressed as:

$$N_i(x,\Omega) = \sum_{j=1}^{m} v_{ij}\varphi(z_{ij})\ ,\ \forall\ i = 1,2,\dots,n \quad \dots(7)$$

where $z_{ij} = w_{ij}x + u_{ij}$, w_{ij} represents the weight from the input to the j^{th} hidden unit, v_{ij} represents the weight from the j^{th} hidden unit to the output unit, and u_{ij} represents the bias for the j^{th} hidden node.

In this investigation we have considered the sigmoid function $\varphi(x) = \frac{1}{1+e^{-x}}$ as an activation function.

The approximate solution $u_{iN}(x,\Omega)$ satisfies the initial conditions and the error function may be computed as follows:

$$E_i(x_l,\Omega) = CD_a^{\alpha_i(x_l)}u_{iN}(x_l,\Omega) - \sum_{i=1}^{n}\int_0^{x_l} k_i(x_l,t)\,u_{iN}(t,\Omega)dt - f_i(x_l)\ , \forall i = 1,2,\dots,n-1 \quad \dots(8)$$

and

$$E_n(x_l,\Omega) = g(x_l, u_{1N}(x_l,\Omega), u_{2N}(x_l,\Omega), \dots, u_{nN}(x_l,\Omega)) \quad \dots(9)$$

where x_l , $l = 1,2,\dots,h$ are collocation points in (0, T].

We now construct an unconstrained minimization problem related to the system (1)-(3) as:

$$\text{Minimize } \sum_{l=1}^{h}\sum_{i=1}^{n-1}[E_i(x_l,\Omega)]^2 + \sum_{l}^{h}[E_n(x_l,\Omega)]^2 \quad \dots(10)$$

The minimization problem (10) can be written as follows

$$\text{Minimize } E(x,\Omega) = \frac{1}{2}\parallel\ \psi(x,\Omega)\ \parallel_2^2 \quad \dots(11)$$

where:

$$\psi(x,\Omega) = [E_1(x_1,\Omega), E_1(x_2,\Omega), \dots, E_1(x_l,\Omega), E_2(x_1,\Omega), E_2(x_2,\Omega), \dots, E_2(x_l,\Omega), \dots, E_n(x_l,\Omega)]^T$$

4. Error Backpropagation Learning Algorithm (EBPLA)

The EBPLA will be used to crush the network constraints (weights) and for minimizing the total error function $E(x,\Omega)$ of the ANN. Here the gradient descent method [42] has been used for modifying the parameters.

$$w_{ij}^{k+1} = w_{ij}^k + \Delta w_{ij}^k = w_{ij}^k + \left(-\eta\frac{\partial E(x,\Omega)^k}{\partial w_{ij}^k}\right) \quad \dots(12)$$

$$v_{ij}^{k+1} = v_{ij}^k + \Delta v_{ij}^k = v_{ij}^k + \left(-\eta\frac{\partial E(x,\Omega)^k}{\partial v_{ij}^k}\right) \quad \dots(13)$$

where η is the learning parameter, k is an iteration step that is used to update the weights, and $E(x,\Omega)$ is the total error function.

5. Structure of Multi-Layer ANN Model for VFIDAEs

We consider a three-layer ANN model for the present problem given by equations (1)-(3). The construction of neural network architecture, which consists of an input layer with single input node and a bias, one hidden layer having five hidden nodes and output layer contains one output node. Initial weights w_j from input to hidden layer and v_j from hidden to output layer are considered as random. Architecture of the three layers ANN with five hidden nodes, single input and output layer (with one node) is shown in Figure 1.

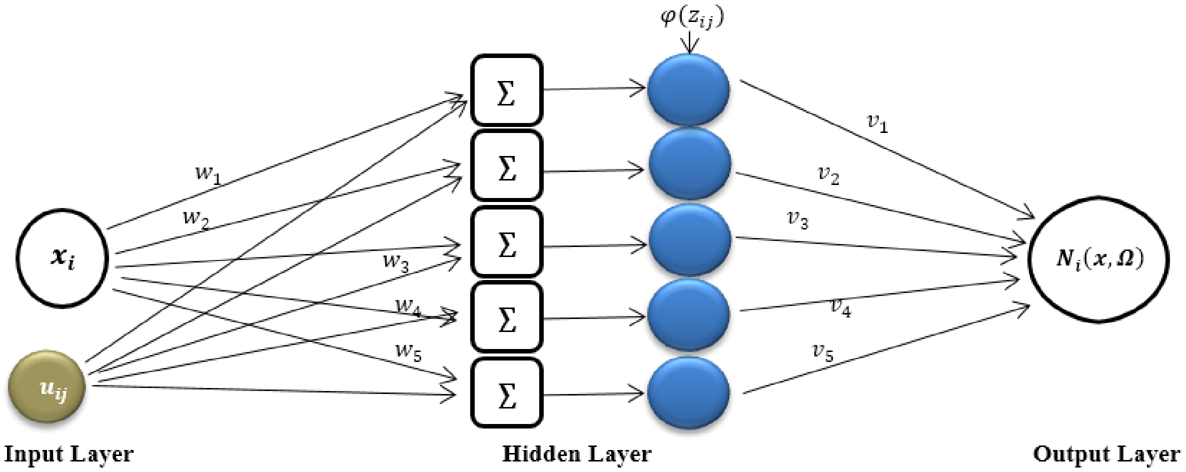

Figure 1. Proposed ANN architecture.

6. Computation of the Gradient for VFIDAEs

For minimizing the error function $E(x,\Omega)$ that is to update the network parameters (weights), we differentiate $E(x,\Omega)$ with respect to the parameters. Thus, the gradient of network output with respect to their inputs is calculated as follows.

The conformable variable order fractional derivative of $N_i(x,\Omega)$ is given by:

$$CD_a^{\alpha_i(x)} N_i(x,\Omega) = v_{ij}\varphi'(z_{ij})w_{ij}x^{1-\alpha_i(x)} \qquad \ldots\textbf{(14)}$$

Let $CD_a^{\alpha_i(x)} N_i(x,\Omega) = N_{i\beta}$ represents the derivative of the network output with respect to its inputs. The derivative of $N_{i\beta}$ with respect to other parameters may be found as (according to conformable variable order fractional derivative rules)

$$\frac{\partial N_{i\beta}}{\partial w_{ij}} = v_{ij}x^{1-\alpha_i(x)}\left(\varphi'(z_{ij}) + \varphi''(z_{ij})w_{ij}x\right) \qquad \ldots(15)$$

$$\frac{\partial N_{i\beta}}{\partial v_{ij}} = w_{ij}x^{1-\alpha_i(x)}\varphi'(z_{ij}) \qquad \ldots(16)$$

$$\frac{\partial N_{i\beta}}{\partial u_{ij}} = w_{ij}v_{ij}x^{1-\alpha_i(x)}\varphi''(z_{ij}) \qquad \ldots(17)$$

Here:

$$N_i(x,\Omega) = \sum_{j=1}^{m} v_{ij}\varphi(z_{ij}) \text{ and } z_{ij} = w_{ij}x + u_{ij}$$

From (6) we have (by differentiating)

$$CD_a^{\alpha_i(x)} u_{iN}(x) = (x-x_0)^{1-\alpha_i(x)}N_i(x,\Omega) + (x-x_0)CD_a^{\alpha_i(x)}N_i(x,\Omega)) \qquad \ldots(18)$$

After simplifying the above equation, we get

$$CD_a^{\alpha_i(x)} u_{iN}(x) = (x-x_0)^{1-\alpha_i(x)}N_i(x,\Omega) + (x-x_0)(w_{ij}v_{ij}\varphi'(z_{ij})x^{1-\alpha_i(x)}) \qquad \ldots(19)$$

7. Algorithm

Step 1. Randomly select the initial values of adjustable parameters $\Omega_s, s = 1,2,\ldots,n(nm)$ and select an error tolerance parameter $\varepsilon > 0$.

Step 2. Initialize the input vector $x = [x_1, x_2, \ldots, x_l]$.

Step 3. Calculate the output values $N_i(x,\Omega)$, $i = 1,2,\ldots,n$ using equation (7).

Step 4. Calculate the error function $E(x,\Omega)$ using equation (11).

Step 5: Update the parameters using unsupervised backpropagation algorithm $\Omega_s^{k+1} = \Omega_s^k + \left(-\eta \frac{\partial E(x)^k}{\partial \Omega_s^k}\right)$.

Step 6: If $E(x,\Omega) \le \varepsilon$, then go to step 7 else go to step 2.

Step 7: The last parameters will be stored after the learning algorithm finished.

The diagram of the constructing the learning algorithm for the proposed approach is shown in Figure 2.

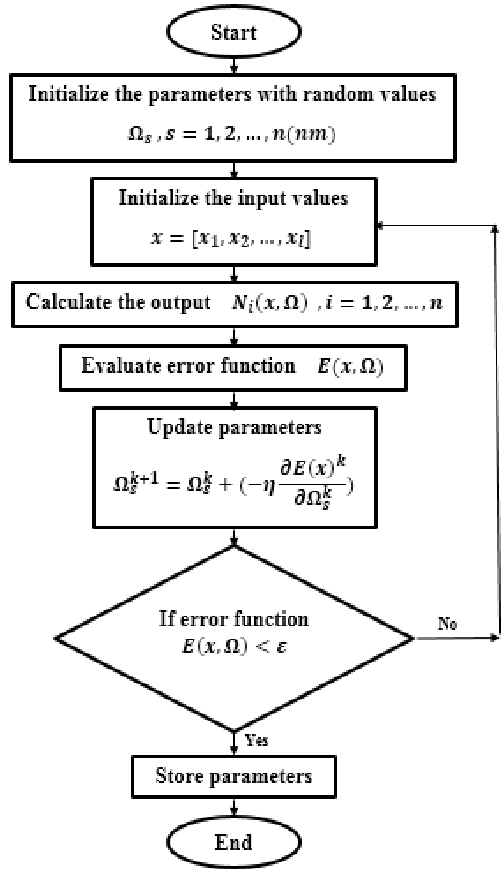

Figure 2. Diagram of the constructing the learning algorithm for the approach.

8. Numerical Examples

The capability of the approach described in the preceding part was demonstrated by applying it to some tested examples.

The approximate results by ANN model are compared with exact solutions of each example in order to show the powerfulness of the proposed method.

Example 8.1. Consider the following VFIDAEs:

$$CD_a^{\alpha(x)}u(x) = x\int_0^1 u(t)dt + \int_0^1 v(t)dt + f(x) \quad \ldots(20)$$

$$0 = u(x) + v(x) - e^{-x} - \sin x \quad \ldots(21)$$

The function $f(x)$ is elected such that the closed form of the problem is $u(x) = e^{-x}$, $v(x) = \sin x$ and the initial conditions $u(0) = 1, v(0) = 0$.

The corresponding ANN approximate solution is expressed as:

$$u_N(x,\Omega) = 1 + xN_1(x,\Omega)$$

and

$$v_N(x,\Omega) = xN_2(x,\Omega)$$

Next, we construct the error functions:

$$E_1(x,\Omega) = CD_a^{\alpha(x)}u_n(x,\Omega) - x\int_0^1 u_n(t,\Omega)dt - \int_0^1 v_N(t,\Omega)dt - f(x) \quad \ldots(22)$$

and

$$E_2(x,\Omega) = u_n(x,\Omega) + v_N(t,\Omega) - e^{-x} - \sin x \quad \ldots(23)$$

Therefore

$$E(x,\Omega) = \frac{1}{2}\sum_{i=1}^{10}\{(E_1(x_i,\Omega))^2 + \left(E_2(x_i,\Omega)\right)^2\} \quad \ldots(24)$$

The network is trained for ten equidistant points in [0, 1] and five hidden nodes. After more than 30000 iterations we get the ANN solutions. Comparison between exact and ANN solutions for $\alpha(x)=0.5$, $\alpha(x)=1-\frac{x}{2}$ and $\alpha(x)=0.97-0.03\cos\left(\frac{x}{10}\right)$ are cited in Figures 3-5 respectively. The values of the maximum error and the mean squared errors for some different values of $\alpha(x)$ are listed in Table 1. One may see that the approximate solutions from the proposed ANN method have excellent agreement with the exact solutions for different values of $\alpha(x)$.

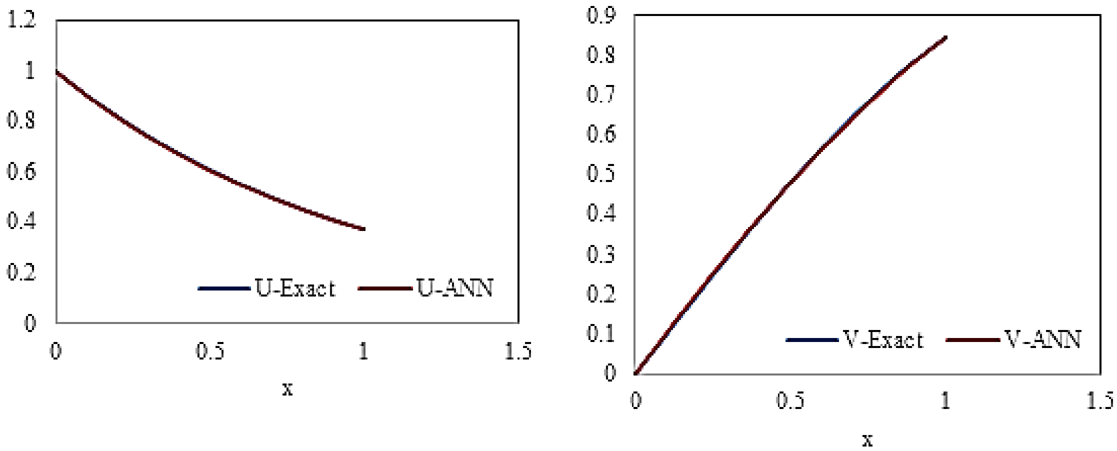

Figure 3. Exact and ANN results for $\alpha(x)=0.5$.

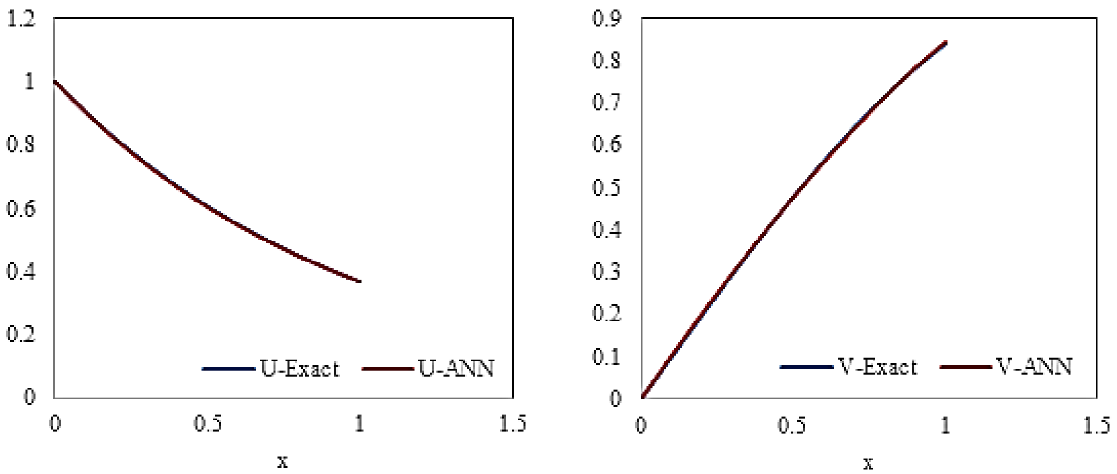

Figure 4. Exact and ANN results for $\alpha(x)=1-\frac{x}{2}$.

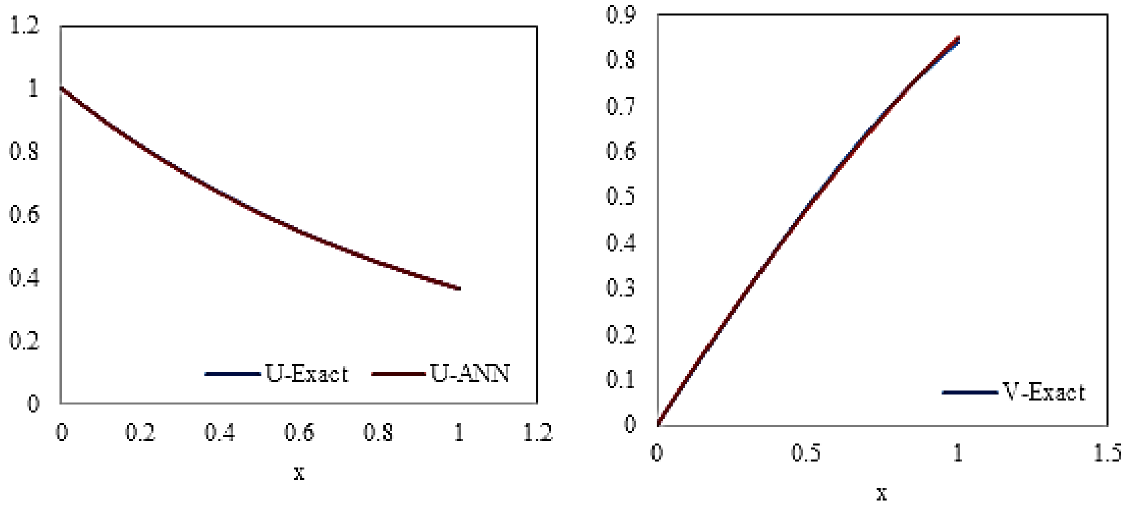

Figure 5. Exact and ANN results for $\alpha\,(x)=0.97-\cos\frac{x}{10}$.

Table 1. The maximum error and the MSE of Example 8.1.

α(x)	**Max. Error**	**MSE of *U***	**MSE of *V***
0.5	3.499982×10^{-4}	3.7137×10^{-6}	1.5248×10^{-5}
$1-\frac{x}{2}$	4.999940×10^{-4}	3.0907×10^{-6}	1.6361×10^{-5}
$0.97-\cos\frac{x}{10}$	9.998299×10^{-4}	3.1004×10^{-6}	3.4722×10^{-5}

Example 8.2. Consider the following VFIDAEs:

$$CD_a^{\alpha(x)}u(x) = \int_0^1 u(t)dt + \int_0^1 v(t)dt + f(x) \quad \ldots(25)$$

$$(u(x))^2 + \frac{1}{2}v(x) = e^{-2x} + \frac{\sin x}{2} \quad \ldots(26)$$

With exact solution $u(x) = e^{-x}$, $v(x) = \sin x$, and initial condition $u(0) = 1$, $v(0) = 0$.

Suppose:

$u_N(x,\Omega) = 1 + xN_1(x,\Omega)$, $v_N(x,\Omega) = xN_2(x,\Omega)$

So:

$$E_1(x,\Omega) = CD_a^{\alpha(x)}u_n(x,\Omega) - \int_0^1 u_n(t,\Omega)dt - \int_0^1 v_N(t,\Omega)dt - f(x) \quad \ldots(27)$$

and

$$E_2(x,\Omega) = (u_n(x,\Omega))^2 + \frac{1}{2}v_N(t,\Omega) - e^{-2x} - \frac{1}{2}\sin x \quad \ldots(28)$$

Hence:

$$E(x,\Omega) = \frac{1}{2}\sum_{i=1}^{10}\{(E_1(x_i,\Omega))^2 + \left(E_2(x_i,\Omega)\right)^2\} \quad \ldots(29)$$

The ANN solution is obtained after more than 40000 iterations. Comparison between the exact and ANN solutions for $\alpha\,(x)=0.5$, $\alpha\,(x)=1-\frac{x}{2}$ and $\alpha\,(x)=0.97-0.03\cos\left(\frac{x}{10}\right)$ are cited in Figures 6-8 respectively. The values of the maximum error and the mean squared errors for some different values of $\alpha(x)$ are listed in Table 2.

ANJS, Vol.25 (3), September, 2022, pp. 25-32

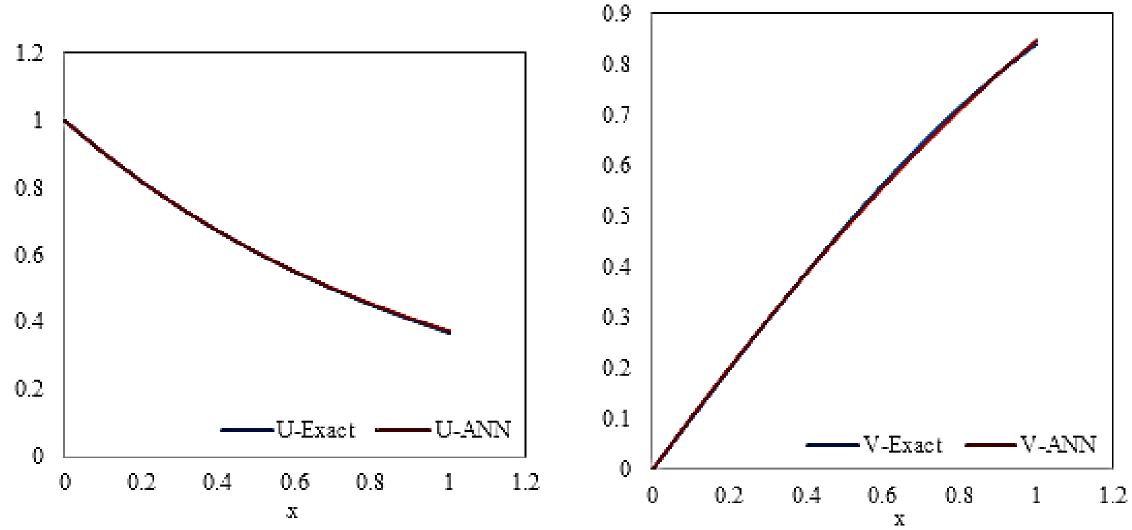

Figure 6. Exact and ANN results for $\alpha(x) = 0.5$.

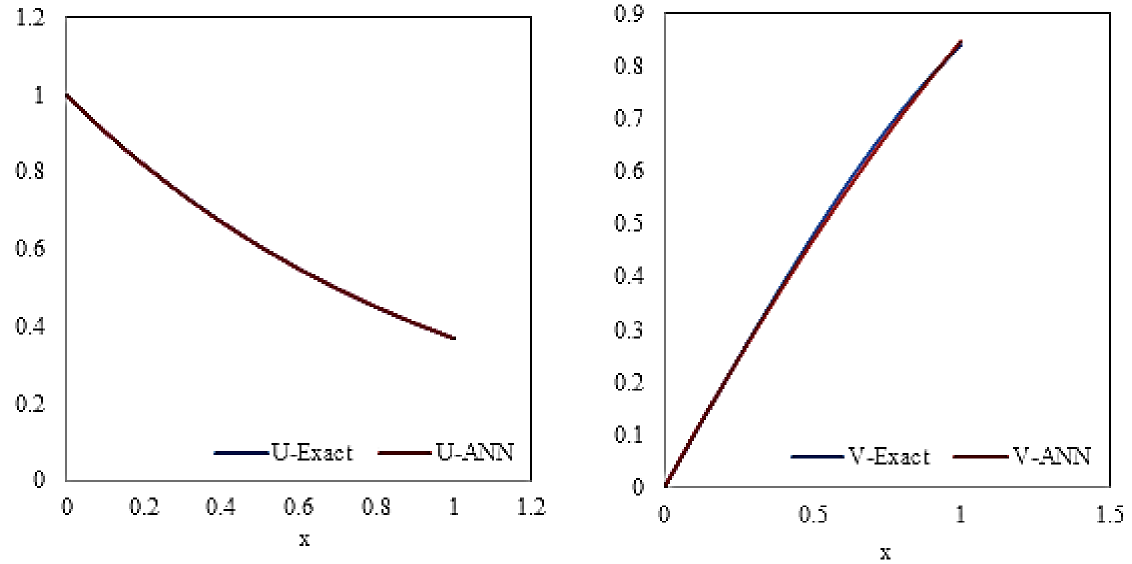

Figure 7. Exact and ANN results for $\alpha(x) = 1 - \frac{x}{2}$.

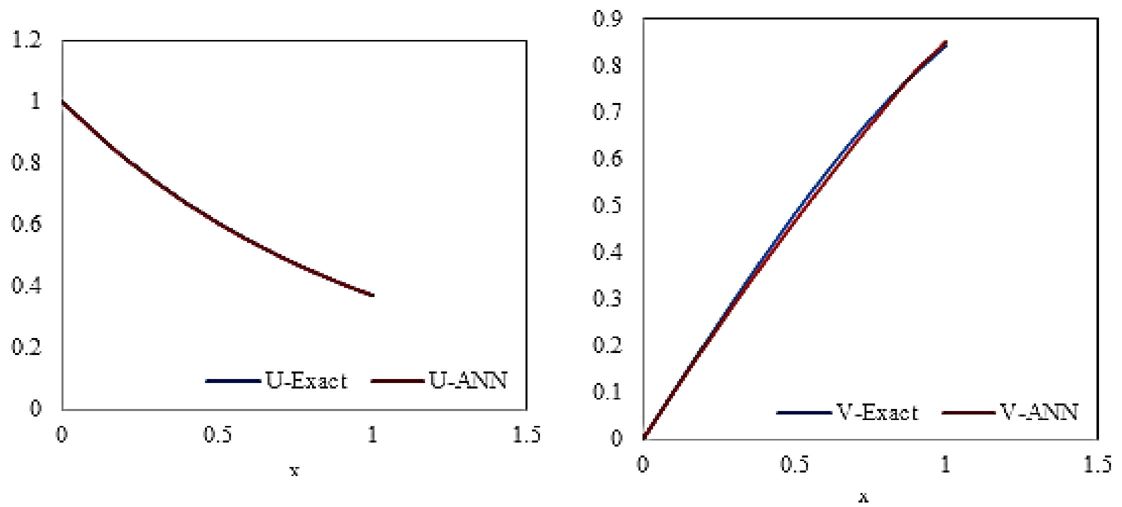

Figure 8. Exact and ANN results for $\alpha(x) = 0.97 - \cos\frac{x}{10}$.

Table 2. The maximum error and the MSE of Example 8.2.

α(x)	**Max. Error**	**MSE of U**	**MSE of V**
0.5	9.999190×10^{-5}	6.2693×10^{-6}	2.9309×10^{-5}
$1-\frac{x}{2}$	9.999381×10^{-5}	1.7025×10^{-5}	6.9724×10^{-5}
$0.97-\cos\frac{x}{10}$	4.999943×10^{-4}	9.8193×10^{-6}	1.3141×10^{-4}

9. Conclusions

This paper presents a new approach to solve VFIDAEs by using ANN model. Correctness of the proposed method has been examined by solving various VFIDAEs. Moreover, the algorithm is unsupervised and error backpropagation algorithm is used to minimize the error function. Consistent initial weights from input to hidden and from hidden to output are random. One may see from the graphs that the approximate solutions by ANN is effective. Finally, the ANN method is both good and simple in terms of computing.

References

[1] Laroche E. and Knittel D.; “An improved linear fractional model for robustness analysis of a winding system" Control Engineering Practice, 13(5): 659-666, 2005.

[2] Trujillo J. J.; Scalas E.; Diethelm K. and Baleanu D.; “Fractional calculus: models and numerical methods”. 5. World Scientific, 2016.

[3] Sabatier J.; Aoun M.; Oustaloup A.; Gregoire G.; Ragot F. and Roy P.; “Fractional system identification for lead acid battery state of charge estimation”. Signal processing, 86(10): 2645-2657, 2006.

[4] Uchaikin V. V.; “Fractional derivatives for physicists and engineers”. Volume 2. Springer, 2013.

[5] Atanackovi´c T.; Pilipovi´c S.; Stankovi´c B. and Zorica D.; “Fractional calculus with applications in mechanics: vibrations and diffusion processes”. Wiley, 2014.

[6] Li X.; “Numerical solution of fractional differential equations using cubic B-spline wavelet collocation method”. Communications in Nonlinear Science and Numerical Simulation, 17(10): 3934-3946, 2012

[7] Podlubny I.; “Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications”. Elsevier, 1998.

[8] Ciesielski M. and Leszczynski J.; “Numerical simulations of anomalous diffusion”. arXiv preprint math-ph/0309007, 2003.

[9] Odibat Z.; “Approximations of fractional integrals and Caputo fractional derivatives”. Applied Mathematics and Computation,178(2): 527-533, 2006.

[10] Momani S. and Odibat Z.; “Numerical approach to differential equations of fractional order”. Journal of Computational and Applied Mathematics, 207(1): 96-110, 2007.

[11] Momani S. and Odibat Z.; “Analytical solution of a time-fractional Navier–Stokes equation by Adomian decomposition method”. Applied Mathematics and Computation, 177(2): 488-494, 2006.

[12] Jafari H.; Yousefi S.; Firoozjaee M.; Momani S. and Khalique C. M.; “Application of Legendre wavelets for solving fractional differential equations”. Computers & Mathematics with Applications, 62(3): 1038-1045, 2011.

[13] Wu G.-c. and Lee E.; “Fractional variational iteration method and its application”. Physics Letters A, 374(25): 2506-2509, 2010.

[14] Das S.; “Analytical solution of a fractional diffusion equation by variational iteration method”. Computers & Mathematics with Applications, 57(3): 483-487, 2009.

[15] Salahshour S.; Ahmadian A.; Senu N.; Baleanu D. and Agarwal P.; “On analytical solutions of the fractional differential equation with uncertainty: application to the basset problem”. Entropy, 17(2): 885-902, 2015.

[16] Salahshour S.; Allahviranloo T. and Abbasbandy S.; “Solving fuzzy fractional differential equations by fuzzy laplace transforms”. Communications in Nonlinear Science and Numerical Simulation, 17(3): 1372-1381, 2012.

[17] Kadhem M. F. and Alfayadh A. H.; “Mixed Homotopy Integral Transform Method for Solving Non-Linear Integro-Differential Equation" Al-Nahrain Journal of Science 25(1): 35-40, 2022.

[18] Ganji R. and Jafari H.; “Numerical solution of variable order integro-differential equations”. Advanced Mathematical Models & Applications, 4(1) :64-69, 2019.

[19] Zurada J. “Introduction to artificial neural systems, west pub”. Co.; St. Paul, MN, 1992.

[20] Graupe D. “Principles of artificial neural networks”. volume 7. World Scientific, 2013.

[21] Hanss M.; “Applied fuzzy arithmetic”. Springer, 2005.

[22] Wang G. and Liang J.; “Exponential stability analysis for delayed stochastic Cohen–Grossberg neural network”. International Journal of Computational Intelligence Systems, 3(1): 96-102, 2010.

[23] Shen X.; Gao X. Z. and Bie R.; “Artificial immune networks: Models and applications”. International Journal of Computational Intelligence Systems, 1(2): 168-176, 2008.

[24] S Mall S. and Chakraverty S.; “Application of Legendre Neural Network for solving ordinary

differential equations". Applied Soft Computing, 43: 347-356, 2016.
[25] Mall S. and Chakraverty S.; "Hermite functional link neural network for solving the van der Pol–Duffing oscillator equation". Neural computation, 28(8): 1574-1598, 2016.
[26] Mall S. and Chakraverty S.; "Numerical solution of nonlinear singular initial value problems of Emden–Fowler type using Chebyshev Neural Network method". Neurocomputing, 149: 975-982, 2015.
[27] Mall S. and Chakraverty S.; "Chebyshev Neural Network based model for solving Lane-Emden type equations". Applied Mathematics and Computation, 247: 100-114, 2014.
[28] Chakraverty S. and Mall S.; "Regression based weight generation algorithm in neural network for solution of initial and boundary value problems". Neural Computing and Applications, 25(3): 585-594, 2014.
[29] Chakraverty S. and Mall S.; "Artificial neural networks for engineers and scientists: solving ordinary differential equations". CRC Press, 2017.
[30] Jumarhon B.; Lamb W.; McKee S. and Tang T. "A Volterra integral type method for solving a class of nonlinear initial-boundary value problems". Numerical Methods for Partial Differential Equations: An International Journal, 12(2): 265-281, 1996.
[31] Kafarov V.; Mayorga B. and Dallos C.; "Mathematical method for analysis of dynamic processes in chemical reactors". Chemical engineering science, 54(20): 4669-4678, 1999.
[32] Von L. Wolfersdorf, "On identification of memory kernels in linear theory of heat conduction". Mathematical methods in the applied sciences, 17(12): 919-932, 1994.
[33] Bellen A.; "Collocation methods for Volterra Integral and Related Functional Equations". 2004, Cambridge University Press. Apparsa in JIEA, 2005.
[34] Gomilko A.; "A Dirichlet problem for the biharmonic equation in a semi-infinite strip". Journal of engineering mathematics, 46(3): 253-268, 2003.
[35] Pishbin S.; Ghoreishi F. and Hadizadeh M.; "A posteriori error estimation for the legendre collocation method applied to integral-algebraic Volterra equations". Electron. Trans. Numer. Anal, 38: 327-346, 2011.
[36] Pishbin S.; Ghoreishi F. and Hadizadeh M.; "The semi-explicit Volterra integral algebraic equations with weakly singular kernels: the numerical treatments". Journal of Computational and Applied Mathematics, 245: 121-132, 2013.
[37] Zenchuk A.; "Combination of inverse spectral transform method and method of characteristics: Deformed Pohlmeyer equation. Journal of Nonlinear Mathematical Physics, 15(sup3): 437-448, 2008.
[38] Tavares D.; Almeida R. and Torres D. F.; "Caputo derivatives of fractional variable order: numerical approximations". Communications in Nonlinear Science and Numerical Simulation, 35: 69-87, 2016.
[39] Odzijewicz T.; Malinowska A. B. and Torres D. F.; "Fractional variational calculus of variable order". In: Advances in harmonic analysis and operator theory. Springer, 291-301, 2013.
[40] Zhang S.; Li S. and Hu L.; "The existences and uniqueness result of solutions to initial value problems of nonlinear diffusion equations involving with the conformable variable derivative". Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales. Serie A. Matemáticas, 113(2): 1601-1623, 2019.
[41] Wang J. and Zhang S.; "The existence of solutions for nonlinear fractional boundary value problem and its Lyapunov-type inequality involving conformable variable-order derivative". Journal of Inequalities and Applications, 2020(1): 1-12, 2020.
[42] Wei Y.; Kang Y.; Yin W. and Wang Y.; "Generalization of the gradient method with fractional order gradient direction". Journal of the Franklin Institute, 357(4): 2514-2532, 2020.