Progress in Fractional Differentiation and Applications
An International Journal

http://dx.doi.org/10.18576/pfda/050306

On the Solving Fractional Volterra-Type Differential Equations by Using Artificial Neural Networks Approach

Ahmad Jafarian[1], *Fariba Rostami* [1] *and Alireza Khalili Golmankhaneh*[2,*]

[1] Department of Mathematics, Urmia Branch, Islamic Azad University, Urmia, Iran
[2] Young Researchers and Elite Club, Islamic Azad University, Urmia Branch,Urmia, Iran

Received: 2 Jul. 2018, Revised: 10 Sep. 2018, Accepted: 14 Sep. 2018
Published online: 1 Jul. 2019

Abstract: Lately, there is a great concern in the applications of the artificial neural networks approach in modeling and mathematically analyze various complex real-world phenomena. In this literature, one of the most successful and effective neural network architectures has been implemented to construct the numerical solution of the fractional Volterra-type equations. For this aim, one supervised back-propagation type learning algorithm which is planned on a three-layered feed-forward neural network is applied for approximating the mentioned problem as a convergent power series solution. To be more precise, we have also considered some numerical examples with the comparison to the results given by the Euler wavelet method. Obtained simulation and numerical results illustrate that the proposed iterative technique is globally convergent and specially efficient for solving this fractional problem.

Keywords: Non-linear fractional order Volterra integro-differential equation, power-series method, artificial neural networks approach, criterion function, back-propagation learning algorithm.

1 Introduction

As it is well known, one priority and central area of modern mathematics is solving the integro-differential equations (IDEs), which have an important role in the description of more and more phenomena in the world around us. As a matter of fact, finding the exact solution of such problems is not available in most of the cases, therefore some non-analytical methods should be constructed according to the needs. There are many different scientific papers and primary research articles devoted to the construction and investigation of some numerical solutions with good performance of convergence to the fractional differential equations (FDEs). For example, Belbas in [1] studied the extension of Hill's method of infinite determinants to the case of IDEs with periodic coefficients and kernels. The modified Laplace Adomian decomposition method was employed by Manafianheris in [2], to approximate solutions of these equations. In [3], ameliorated Legendre collocation method was given for a class of IDEs which include a population model. It is to be noted that the present technique was built by utilizing the residual function of the operator equation. As the first time, Babaaghaie et al. in [4] provided a suitable wavelet Galerkin method in the Holder space for Fredholm type IDEs. The numerical solutions of delay IDEs were studied by modification of continuous collocation Taylor polynomials [5]. The analytical approximate method have been used to solve differential equation [6-9]. In [10], the authors developed the double exponential transformation technique which is based on the Sinc-collocation method for the numerical solution of IDEs. In [11], a parametric iteration method was purposed by Tamamgar to the numerical solution of Fredholm-type IDEs. The quadratures method was mainly used in [12] for obtaining the approximate solutions of that equation. In the above mentioned references, the authors suggested a number of numerical methods which were used to solve a variety of IDEs (see also the references therein). In this article, we would like to continue our work to study a special type of differential equations, namely initial valued non-linear fractional Volterra-type equations. So far, more work has been done on the fractional calculus models in the case of fractional differential equation problem [13,14,15]. The most important and numerous cases of these works are in [16,17]. In [18,19], two modifications of hybrid collocation method were used to solve the indicated fractional problem. Taheri et al. in [20] used the shifted Legendre spectral collocation method to solve the stochastic

* Corresponding author e-mail: alirezakhalili2002@yahoo.co.in

fractional integro-differential equations (FIDE). The main characteristic of this method is reducing the original problem into a system of algebraic equations which could be solved with Newton's method. In [21], the least squares method with aid of shifted Chebyshev polynomial was applied to the numerical solution of linear FIDE. In [22], Oyedepo et al. have employed the standard and perturbed least squares method involving Bernstein polynomials function for solving FIDEs. The presented technique reduces the mentioned fractional problem into a system of linear algebraic equations which can be solved by a standard numerical method. Shahmorad et al. in [23] introduced a numerical approach which preserves the geometric structure on the Lorentz Lie group to the numerical solution of a FIDE boundary value problem. In the present technique, under a semi-discretization method and use of a Newton-Cotes quadrature rule the original equation was converted to a system of ordinary differential and the obtained system was solved by the group- preserving scheme. Wang and Zhu in [24] used the wavelet numerical method which was based upon Euler wavelet approximations to solve the non-linear Volterra FIDEs. By applying this technique, the fractional problem was reduced to a system of algebraic equations which could be solved through a known standard numerical method. Unlike at papers investigated, the main aim of this paper is to offer an iterative scheme to approximate solution of an initial value problem for fractional Volterra IDE by the Artificial Neural Network (ANNs) approach. Several structures of ANNs have been employed to the numerical solution of fractional problems due to the complexity of mathematical problems derived from functional models [25-27]. The ANN combined with genetic algorithm, particle swarm optimization, active-set method, sequential quadratic programming, interior-point algorithm and their hybrids have been used for solving differential equation [28-31]. As known, the ANNs approach is a powerful mathematical framework which can be successfully used in a wide variety of applications (For more details see [32] and the references therein). Here, we aim to apply this computing paradigm to approximate solution of a non-linear fractional Volterra-type equation. For this reason, at first a particular three-layered feed-forward neural architecture is designed equivalently to the mathematical problem to completely model that. In this manner, the differential domain is partitioned into some subintervals to define a set of training patterns. Now, we train the designed neural architecture with the use of feeding these data set, and let it change its parameters (connection weights and bias terms) according to a learning algorithm. In other words, we train the network by modifying the random-valued bias terms and weights of connections between the nodes of the hidden and output layers. It should be noted here that the training begins with random weights and biases, and the goal is to adjust the network parameters by using the back-propagation learning algorithm, so that the network error will be minimal.
Outline of the paper is as follows:
Section 2 begins with introducing some fundamental concepts of the artificial neural networks and fractional calculus. In the following, we outline how to construct our referred method to approximate solution of the objective non-linear initial value problem. Some numerical examples comparing the solutions and error terms with previously-known methods are provided in section 3. Finally, conclusion is given in the last section of the paper.

2 Description of the method

The main feature of this section is to offer a combination iterative technique to the numerical solution of an initial valued non-linear fractional Volterra type equation. The solution of the mathematical problem is achieved by first defining an initial approximation or trial solution. Then, a highly precise representation will be obtained employing the artificial neural networks approach. To illustrate basic concepts of the method, we first start with understanding formulation of considered neural architecture and then introducing mathematical thoughts of fractional calculus theory.

2.1 Basic idea of ANNs

Human being has learned a lot from nature. In substance, one has applied as some patterns as possible. Besides, one seeks now to map out some of the perceptual and intelligence characteristics of the human brain. Neural network is a revelation that tries to imitate the processing power of human brain. This method of machine learning can be used to study data in massive parallel processing. It provides data-parallel implementations of machine learning. Interest in studying the various commercial applications of the neural network has grown since the advent of computer technology in 1980s. The findings of most researchers indicate that neural network technology has been able to successfully deal with business issues and is, in most cases, superior to other traditional technologies. Neural networks have a really high ability to deduce results from vague and complex data in pattern extraction, and are highly effective in identifying methods that are not easy to be understood by human's knowledge and of other computer-based techniques. A trained neural network can act as an expert in the category given for analysis, and provide estimations on desired situations and future predictions. The structure of neural networks is such that it is capable of solving the problem without the help of an expert and identifies patterns which are out of data that nobody knows about their existence. For a better

understanding of topics related to neural networks, one can be referred to these references [33-35].
To practice the theoretical issues, we have provided the multi-layer feed-forward neural network shown in Figure 1. The given structure is composed of artificial neurons which are arranged in the three different layers. The patterns of information are imputed from the left-hand side layer and outputted to the right hand one. The first column from the input side is called the input layer, the second is considered as the hidden layer and the last one is named the output layer. The input units receive various signals from the outside world, and then feed them into the hidden layer via a connection set which are called weight parameters. These signals trigger the hidden layer, and then are multiplied by their corresponding weight parameters. The bias terms are added to every neuron in the both hidden and output layers, before the activation function is used. The output unit sums all of its input signals to provide the network output. Here, the identity activation function is used throughout the designed neural network. According to the above descriptions, the input-output mathematical relations corresponding to each neuron are summarized below:

• *Input unit:*

$$o^1 = y. \tag{1}$$

• *Hidden units:*

$$o_i^2 = f(net_i), \tag{2}$$
$$net_i = y.v_i,\ i = 1,...,n.$$

• *Output unit:*

$$N(y) = \sum_{i=1}^{n} (o_i^2.w_i) + b. \tag{3}$$

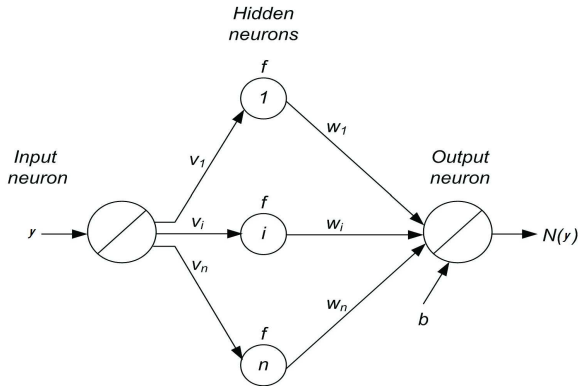

Fig. 1. *Block diagram of the represented neural architecture.*

The designed neural network contains some learning patterns which modify the connection weights and bias terms according to a standard mathematical process. In this way, the supervised back-propagation learning algorithm is employed to make an appropriate adjustments to the network parameters. In the following sections, it will be further explained how this neural architecture is able to actually approximate the mathematical problem.

2.2 Implementation of the method

It is of crucial importance, one could obtain solutions for the fractional differential equation. However, drastic scientific shifts in research problems are provoked once in a while in the evolution of science. It is a system of development that includes a scientific field progress in the theory of chaos and fractals that pointed out relationships between fractional derivatives and integrals, leading to create interest in this field. Mathematicians such as Euler and Liouville have, indeed, expanded their vision to the calculation of non-integer order derivatives and integrals. Fractional order calculus, following important shifts perceived in the field, tries to witness the process with high-order dynamics and complex non-linear phenomena using few coefficients. More simply said, the arbitrary order of the derivatives creates an additional degree of freedom to share a specific behavior. It would be meaningful to claim here that fractional order derivatives proceeds through the whole characteristic of the function in detail. That is, any evaluation point would be needed depending on the past values of the function in case the system has a long-term ”memory”. With these ideas in mind, this work introduces the fundamentals of fractional order calculus and its applications. Within this perspective, setting certain principles to the theory, some other researchers have become interested in examining this new possibility

in other scientific areas. For more details, we refer to [16,17,36]. We continue this paper by presenting the basic definitions and properties of fractional order derivative and integral that are frequently used [16,17]. As is well known, there are several definitions for fractional order integrals and derivatives, such as Riemann-Liouville, Caputo, Grunwald-Letnikov, etc. Here, we present the most famous one, namely the Caputo's definition [16,17]. Fractional differential equations have been used to model processes with memory effect, non-conservative, fractional monopoles, chaotic systems, processes in fractal medium, and anomalous diffusion [16,17,37].

Definition. A function $g(t)$ is continuously differentiable on $[c,d]$ up to order m. The Caputo derivative ${}_aD_t^{\alpha}$ of order $\alpha > 0$ is given by:

$$ {}_cD_t^{\alpha}[g(t)] = \begin{cases} \frac{d^m u(t)}{dt^m} & ,\alpha = m \in N \\ \frac{1}{\Gamma(m-\alpha)} \int_c^t \frac{u^{(m)}(\eta)}{(t-\eta)^{\alpha-m+1}} d\eta, \ t > c & ,0 \le m-1 < \alpha < m \end{cases} . \tag{4} $$

Here, $\Gamma(.)$ symbolizes the Gamma function, and also the following property holds:

$$ {}_cD_t^{\alpha}[t^k] = \begin{cases} 0 & ,m \in Z^+, \ m < \lceil \alpha \rceil \\ \frac{\Gamma(m+1)}{\Gamma(m+1-\alpha)} t^{m-\alpha}, \ t > c & ,m \in Z^+, \ m \ge \lceil \alpha \rceil \end{cases} . \tag{5} $$

Here relations $\lceil \alpha \rceil$ define the smallest integer greater than or equal to constant α. For more details, please review [16,17].

Now, we intend to use the suggested multi-layered feed-forward neural architecture for obtaining the numerical solution of non-linear fractional Volterra-type equation as follows:

$$ P(y).{}_aD_y^{\alpha_1}[u(y)] + Q(y).I_{a,y}^{\alpha_2}[y^{l_1} t^{l_2} \big(u(t)\big)^{l_3}] = H(y), \ 0 < \alpha_1, \alpha_2 \le 1, \tag{6} $$

for the length interval (a,b) and subject to initial condition

$$ u(a) = u_0. $$

In the above differential equation problem, the variable coefficients P, Q and H are deemed continuous real-valued functions.

The area of power series is an interesting mathematical research topic as well as with applications in engineering and computer sciences. On the other hand, the iterative techniques based on artificial neural networks seem to be more appropriate for modeling (and also solving) such complex problems. Hence, we aim to offer an iterative technique of power series method upon the combination and fractional ANNs approach for obtaining a power series solution. Let the solution function $u(y)$ on domain $\Omega = [a,b]$ be represented in a power series of degree I.

$$ u_I(y) = \sum_{i=0}^{I} a_i y^i, \tag{7} $$

be the referred generalized power series for constant coefficients a_i (for $i = 0,...,I$). It should mention here that the solution function $u(y)$ can be represented as the series polynomial (7) if and only if it is complex differentiable in (a,b). For simplicity reasons, this differential interval is replaced with the suitable domain $[0,T]$, which will be used throughout the paper. The relationship between designed neural network and indicated power series are explained further in following parts.

2.3 Discretization of the problems

Consider the following initial trial particular solution:

$$ \tilde{u}(y) = u_0 + y.N(y) = u_0 + y.(\sum_{i=1}^{I} w_i^2 . f(w_i^1 . y) + b), \tag{8} $$

which is used in the remaining parts of this paper. Here, $\tilde{u}(y)$ includes the given feed-forward architecture which is satisfied in the initial condition, simultaneously. Here, the notations $w_i^2 = a_i$, $w_i^1 = x^{i-1}$ and $b = u_0$ are used for corresponding the designed neural network within the power series (7). Substituting $u(y)$ with $\tilde{u}(x)$ leads to the following relation:

$$ P(y).{}_0D_y^{\alpha_1}[u_0 + \sum_{i=1}^{I} a_i y^i] + \tag{9} $$

$$Q(y).I_{0,y}^{\alpha_2}[y^{l_1}t^{l_2}\left(u_0+\sum_{i=1}^{I}a_iy^i\right)^{l_3}]=H(y),\ 0<y<T,\ 0<\alpha_1,\alpha_2\leq 1.$$

Expanding the defined fractional derivative ${}_0D_y^{\alpha_1}$ and fractional integration $I_{0,y}^{\alpha_2}$ on the series including designed neural architecture and then making some algebraic simplification, this leads to the following relation:

$$P(y).\sum_{i=1}^{I}\frac{a_i.\Gamma(i+1).y^{i-\alpha_1}}{\Gamma(i+1-\alpha_1)}+ \tag{10}$$

$$Q(y).I_{0,y}^{\alpha_2}[y^{l_1}.t^{l_2}.(\sum_{k,k',\cdots,k^{(I)}}T_{l_3,k}.T_{k,k'},...,T_{k^{(I-2)},k^{(I-1)}}.u_0^{(I_3-k)}.(a_1.t)^{(k-k')}\times$$

$$(a_2.t^2)^{(k'-k'')}.\cdots.(a_{I-1}.t^{I-1})^{(k^{(I-2)}-k^{(I-1)})}.(a_I.t^I)^{k^{(I-1)}-k^{(I)}})]=H(y),$$

$$\Rightarrow P(y).\sum_{i=1}^{I}\frac{a_i.\Gamma(i+1).y^{i-\alpha_1}}{\Gamma(i+1-\alpha_1)}+$$

$$Q(y).I_{0,y}^{\alpha_2}[y^{l_1}.(\sum_{k,k',\cdots,k^{(I)}}T_{l_3,k}.T_{k,k'},...,T_{k^{(I-2)},k^{(I-1)}}.u_0^{(I_3-k)}.a_1^{(k-k')}\times$$

$$a_2^{(k'-k'')}.\cdots.a_{I-1}^{(k^{(I-2)}-k^{(I-1)})}.a_I^{k^{(I-1)}-k^I}.t^{\frac{(k-k^{(I)}).I.(I+1)}{2}+l_2})]=H(y),$$

$$\Rightarrow P(y).\sum_{i=1}^{I}\frac{a_i.\Gamma(i+1).y^{i-\alpha_1}}{\Gamma(i+1-\alpha_1)}+$$

$$Q(y).[(\sum_{k,k',\cdots,k^{(I)}}T_{l_3,k}.T_{k,k'},...,T_{k^{(I-2)},k^{(I-1)}}.u_0^{(I_3-k)}.a_1^{(k-k')}\times$$

$$a_2^{(k'-k'')}.\cdots.a_{I-1}^{(k^{(I-2)}-k^{(I-1)})}.a_I^{k^{(I-1)}-k^I}.\frac{\Gamma(\frac{(k-k^{(I)}).I.(I+1)}{2}+l_2+1)}{\Gamma(\frac{(k-k^{(I)}).I.(I+1)}{2}+l_2+1+\alpha_2)}\times$$

$$y^{\frac{(k-k^{(I)}).I.(I+1)}{2}+l_1+l_2+\alpha_2})]=H(y),$$

where

$$T_{n,k}=\frac{n!}{n!.(n-k)!},$$

and $k=0,...,l_3;\ k'=0,...,k;\ ...k^{(I)}=0,...,k^{(I-1)}$. Now, a set of acceptable node points should be considered to complete the discretization procedure. For positive integer R, let Ω_r be a partition of domain Ω with the nodal points $y_r=\frac{rT}{R}$, (for $r=0,...,R$). Putting $x=x_r$ and then employing the differentiable least mean square (LMS) function reduces the resulted Eq. (10) into a non-linear optimization problem defined by the following system of equalities:

$$E_r=\frac{1}{2}\Bigg(P(y_r).\sum_{i=1}^{I}\frac{a_i.\Gamma(i+1).y_r^{i-\alpha_1}}{\Gamma(i+1-\alpha_1)}+ \tag{11}$$

$$Q(y_r).[(\sum_{k,k',\cdots,k^{(I)}}T_{l_3,k}.T_{k,k'},...,T_{k^{(I-2)},k^{(I-1)}}.u_0^{(I_3-k)}.a_1^{(k-k')}\times$$

$$a_2^{(k'-k'')}.\cdots.a_{I-1}^{(k^{(I-2)}-k^{(I-1)})}.a_I^{k^{(I-1)}-k^I}.\frac{\Gamma(\frac{(k-k^{(I)}).I.(I+1)}{2}+l_2+1)}{\Gamma(\frac{(k-k^{(I)}).I.(I+1)}{2}+l_2+1+\alpha_2)}\times$$

$$y_r^{\frac{(k-k^{(I)}).I.(I+1)}{2}+l_1+l_2+\alpha_2})]-H(y_r)\Bigg)^2,$$

where

$$T_{n,k}=\frac{n!}{n!(n-k)!},$$

for $0<\alpha_1,\alpha_2\leq 1$ and $r=0,...,R$. Minimizing this cost function by means of the ANNs approach over the space of possible weight parameters is the main objective of the following part. Hence, one suitable error correction learning procedure must be essentially employed to fulfil this goal. More details concerning minimizing techniques can be found in [38].

2.4 Proposed learning algorithm

In scientific studies, the investigations made by the scientists lead to an acceptable fact that ANNs are expected to achieve the same degree of rigor as the human brain deals with learning. ANNs are provided to use their endeavor to adapt with different situations in environment. To do so, an ANN as a complex adaptive system, of course, could access to details, precise, concise and reliable pieces of information passing through it, to change its internal structure. That is, it is capable of changing its input-output behavior upon which environment may be different.Thus, by starting changing the input-output behavior, weights would first be modified and then be adjusted. Certainly, employing learning rules (algorithms or equations) is meant to be necessary. The below proposed algorithm is a supervised error-correcting algorithm for a three layer feed-forward networks with linear activation function. Regarding to this fact that convergence control parameters must be updated, the supervised back-propagation learning algorithm is employed to adjust the initial parameter a_i which has been chosen randomly. The mentioned algorithm is well presented as:

$$a_i(\tau+1) = a_i(\tau) + \Delta a_i(\tau),\ i = 0,...,I, \tag{12}$$

$$\Delta a_i(\tau) = -\eta.D^{\beta}_{a_i}[E_r] + \gamma.\Delta a_i(\tau-1),\ 0 < \beta \leq 1,$$

where η, γ and β are the learning rate, momentum term and fractional order derivative, respectively. In the above relation, the indexes τ and i in $a_i(\tau)$ are ascribed to the iteration number and the label of training connection weight, respectively. Moreover, $a_i(\tau+1)$ and $a_i(\tau)$ depict the adjusted and current weight parameter, respectively. It is true that there is no certain geometric interpretation for fractional derivative of a continuous function at a given point of its domain, but we tend to do this to increase the degree of freedom in the neural network training. Practically, our task is to illustrate that there could be such order to the fractional derivative which by applying it in the procedure, network training will not be time consuming any more and there is going to be less error as well. Here, the fractional order generalized delta learning algorithm which is based on the gradient descent method is used to aid this training process. To complete the derivation of learning procedure, the above fractional order derivative is employed as follows:

$$D^{\beta}_{a_i}[E_r] = \frac{\Gamma(3)}{2.\Gamma(3-\beta)}\Bigg(P(y_r).\frac{a_i.\Gamma(i+1).y_r^{i-\alpha_1}}{\Gamma(i+1-\alpha_1)} +$$

$$Q(y_r).[\Big(\sum_{k,k',\cdots,k^{(I)}} T_{l_3,k}.T_{k,k'},...,T_{k^{(I-2)},k^{(I-1)}}.u_0^{(I_3-k)}.a_1^{(k-k')}\times$$

$$a_2^{(k'-k'')}.\cdots.a_{I-1}^{(k^{(I-2)}-k^{(I-1)})}.a_I^{k^{(I-1)}-k^I}.\frac{\Gamma(\frac{(k-k^{(I)}).I.(I+1)}{2}+l_2+1)}{\Gamma(\frac{(k-k^{(I)}).I.(I+1)}{2}+l_2+1+\alpha_2)}\times$$

$$y_r^{\frac{(k-k^{(I)}).I.(I+1)}{2}+l_1+l_2+\alpha_2}\Big)] - H(y_r)\Bigg)^{2-\beta}\times$$

$$\Bigg(P(y_r).\frac{\Gamma(2).a_i^{(1-\beta)}.\Gamma(i+1).y_r^{i-\alpha_1}}{\Gamma(2-\beta).\Gamma(i+1-\alpha_1)} +$$

$$Q(y_r).[\Big(\sum_{k,k',\cdots,k^{(I)}} T_{l_3,k}.T_{k,k'},...,T_{k^{(I-2)},k^{(I-1)}}.u_0^{(I_3-k)}.a_1^{(k-k')}\times$$

$$a_2^{(k'-k'')}.\cdots.\frac{\Gamma(k^{(i-2)}-k^{(i-1)}+1)}{\Gamma(k^{(i-2)}-k^{(i-1)}+1-\beta)}.a_{i-1}^{(k^{(i-2)}-k^{(i-1)})-\beta}.\cdots.a_{I-1}^{(k^{(I-2)}-k^{(I-1)})}.a_I^{k^{(I-1)}-k^I}\times$$

$$\frac{\Gamma(\frac{(k-k^{(I)}).I.(I+1)}{2}+l_2+1)}{\Gamma(\frac{(k-k^{(I)}).I.(I+1)}{2}+l_2+1+\alpha_2)}.y_r^{\frac{(k-k^{(I)}).I.(I+1)}{2}+l_1+l_2+\alpha_2}\Big)]\Bigg).$$

From the computational relations, it can easily be concluded that modifying the network parameters without the use of a powerful computational software is impossible.

3 Numerical simulation and discussion

In this section, two high-order linear fractional differential equations problems are solved numerically to illustrate the capability of the developed ANNS approach. Besides, comparisons are conducted between the results of this iterative approach and those obtained by other numerical techniques proposed in [24]. Here, the norm-2 of absolute error criterion:

$$E_2 = \left(\frac{1}{R+1} \sum_{r=0}^{R} \left(u(y_r) - \tilde{u}(y_r) \right)^2 \right)^{\frac{1}{2}}$$

is implemented as an index to peruse the overall performance of each method. In the following simulations that illustrate the method discussed in detail, we apply the specifications and standards as follows:

1.Learning rate $\eta = 0.02$,
2.Momentum constant $\gamma = 0.04$,
3.Number of hidden neurons $I = 6$,
4.Number of nodal points $R = 10$.

4 Numerical simulation and discussion

Example 3.1. Consider fractional non-linear Volterra-type equation as follows:

$$ {}_0D_y^{\frac{3}{4}}[u(y)] + I_{0,y}^{1}[y.t.u^4(t)] = H(y),\ 0 \le y < 1,$$

where

$$H(y) = \frac{1}{\Gamma(\frac{1}{4})}(\frac{32}{5}y^{\frac{5}{4}} - 4y^{\frac{1}{4}}) + \frac{1}{10}y^{10} - \frac{4}{3}y^9 + \frac{4}{7}y^8 + \frac{1}{6}y^7,$$

with initial condition $u(0) = 0$ and exact solution $u(y) = y^2 - y$ At first, the weight parameters are quantified with small real- valued random constants. Then, the given differential interval $[0,1]$ is partitioned with the help of regular discretization technique for $R = 11$. Now, the described learning algorithm is implemented on the given training patterns to successively adjust the weight parameters until a suitable solution is found. To demonstrate the accuracy of technique presented in this study, the absolute errors obtained by the fractional ANNs approach (for different values of β) and Euler wavelet method are presented in Table 1.

Table 1. *Numerical results for Example 3.1 ($\tau = 1000$).*

y	*ANN*, $\beta = 1$			*ANN*, $\beta = \frac{\pi}{4}$		
	$I = 4$	$I = 8$	$I = 12$	$I = 4$	$I = 8$	$I = 12$
0	0.1584E-7	0.9047E-8	0.3521E-9	0.3528E-11	0.1301E-11	0.9410E-13
0.1	0.1982E-7	0.8673E-8	0.3171E-9	0.4218E-11	0.1270E-11	0.8397E-13
0.2	0.1844E-7	0.8435E-8	0.3041E-9	0.3594E-11	0.1724E-11	0.8684E-13
0.3	0.2534E-7	0.9147E-8	0.4017E-9	0.4075E-11	0.1075E-11	0.9184E-13
0.4	0.2046E-7	0.9564E-8	0.4219E-8	0.2572E-11	0.1004E-11	0.8751E-13
0.5	0.1007E-7	0.8977E-8	0.3827E-9	0.3718E-11	0.9107E-12	0.8490E-13
0.6	0.1431E-7	0.9017E-8	0.1713E-8	0.2819E-11	0.1424E-11	0.9742E-13
0.7	0.1847E-7	0.8349E-8	0.9931E-9	0.3718E-11	0.2730E-11	0.8480E-13
0.8	0.2184E-7	0.1340E-7	0.6533E-8	0.4140E-11	0.2415E-11	0.1675E-12
0.9	0.1689E-7	0.1045E-7	0.8296E-8	0.3002E-11	0.1841E-11	0.1057E-12

y	$ANN,\ \beta=\frac{\pi}{6}$			$ANN,\ \beta=\frac{\pi}{8}$		
	$I=4$	$I=8$	$I=12$	$I=4$	$I=8$	$I=12$
0	0.6521E-5	0.3521E-6	0.5418E-8	0.6204E-5	0.2715E-6	0.3584E-7
0.1	0.5814E-5	0.2518E-6	0.5517E-8	0.4317E-5	0.2118E-6	0.2943E-7
0.2	0.6237E-5	0.2841E-6	0.4930E-8	0.2843E-5	0.3074E-6	0.3027E-7
0.3	0.4589E-5	0.3084E-6	0.4728E-8	0.1830E-5	0.2248E-6	0.3880E-7
0.4	0.5614E-5	0.1984E-6	0.4617E-8	0.9004E-6	0.9324E-7	0.2987E-7
0.5	0.5037E-5	0.2084E-6	0.5637E-8	0.8808E-6	0.2843E-6	0.2301E-7
0.6	0.6317E-5	0.2471E-6	0.3974E-8	0.8719E-6	0.9388E-7	0.3745E-7
0.7	0.6074E-5	0.2364E-6	0.3877E-8	0.7438E-6	0.9634E-7	0.3061E-7
0.8	0.4982E-5	0.8457E-7	0.4517E-8	0.6925E-6	0.8918E-7	0.2749E-7
0.9	0.5674E-5	0.9025E-7	0.5617E-8	0.5902E-6	0.8017E-7	0.2154E-7

y	$ANN,\ \beta=\frac{\pi}{10}$			$Euler$		
	$I=4$	$I=8$	$I=12$	$I=4$	$I=8$	$I=12$
0	0.4325E-6	0.4367E-8	0.7993E-9	0.1838E-4	0.9700E-6	0.7390E-8
0.1	0.4852E-6	0.2483E-8	0.8604E-9	0.2108E-4	0.1386E-5	0.8643E-8
0.2	0.3961E-6	0.3604E-8	0.9037E-9	0.2380E-4	0.2300E-5	0.9128E-8
0.3	0.7500E-6	0.4505E-8	0.9637E-9	0.2901E-4	0.2851E-5	0.1024E-7
0.4	0.4108E-6	0.3943E-8	0.3444E-8	0.3351E-3	0.3574E-5	0.1963E-7
0.5	0.6573E-6	0.4073E-8	0.8931E-9	0.3894E-4	0.4989E-5	0.2531E-7
0.6	0.1080E-5	0.4175E-8	0.3719E-8	0.4520E-4	0.5948E-5	0.2948E-7
0.7	0.1225E-5	0.5010E-8	0.9271E-9	0.5341E-4	0.6720E-5	0.3327E-7
0.8	0.7834E-6	0.2460E-7	0.2143E-8	0.5980E-4	0.7204E-5	0.4173E-7
0.9	0.1684E-5	0.2045E-7	0.1796E-8	0.6992E-4	0.7911E-5	0.4870E-7

With a little more care in the above results, it can easily be found out that obtained solutions via artificial neural networks approach are more accurate than the Euler one. In practical terms, we also showed that it is possible to use fractional derivative orders in the mentioned learning algorithm.

Example 3.2. Let us consider the following fractional non-linear equation:

$$ {}_0D_y^{\frac{4}{5}}[u(y)] + I_{0,y}^{\frac{1}{2}}[u^2(t)] = \frac{\Gamma(4)}{\Gamma(\frac{16}{5})}y^{\frac{11}{5}} + \frac{\Gamma(7)}{\Gamma(\frac{15}{2})}y^{\frac{13}{2}},\ 0 \le y < 1, $$

subject to the initial condition $u(0)=0$, and exact solution $u(y)=y^3$. Now, the described learning algorithm (12) is implemented to adjust the network parameters. Comparison between the norm-2 of absolute errors is made in Table 2, for $\tau = 1000$. To demonstrate the accuracy of technique presented in this study, the indicated error functions are plotted in Fig. 2, for different values of β.

Table 2. *Numerical results for Example 3.1.*

R	$\beta=\frac{\pi}{6}$	$\beta=\frac{\pi}{4}$	$\beta=\frac{e}{3}$	$\beta=1$
	E_2 *norm*	E_2 *norm*	E_2 *norm*	E_2 *norm*
5	3.0188×10^{-6}	3.1278×10^{-7}	5.4283×10^{-10}	4.5602×10^{-9}
10	9.9716×10^{-7}	1.5484×10^{-7}	2.7472×10^{-10}	2.4518×10^{-9}
15	5.1028×10^{-7}	8.8918×10^{-8}	8.3180×10^{-11}	9.0684×10^{-10}
20	9.9430×10^{-8}	5.1376×10^{-8}	5.8143×10^{-11}	6.7308×10^{-10}
25	7.0088×10^{-8}	2.5627×10^{-8}	2.5390×10^{-11}	4.1879×10^{-10}
30	4.3176×10^{-8}	1.0328×10^{-8}	1.1903×10^{-11}	2.0321×10^{-10}
35	1.4976×10^{-8}	9.3275×10^{-9}	8.5140×10^{-12}	9.8641×10^{-11}
40	8.0873×10^{-9}	8.4822×10^{-9}	5.6517×10^{-12}	5.4577×10^{-11}
45	7.3113×10^{-9}	6.9391×10^{-9}	2.1820×10^{-12}	3.8429×10^{-11}
50	5.0113×10^{-9}	5.4391×10^{-9}	7.1820×10^{-13}	1.8429×10^{-11}

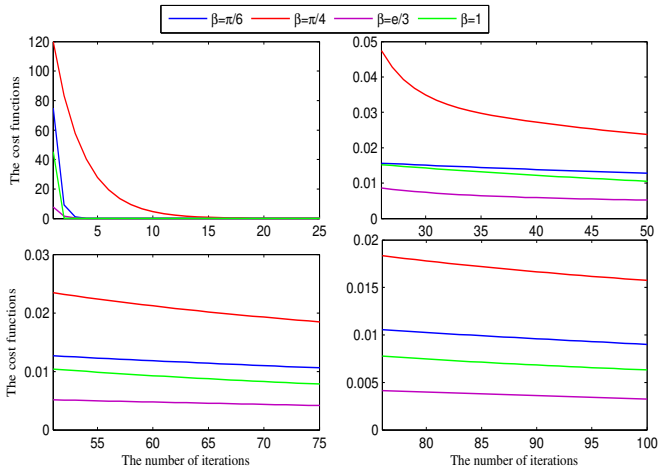

Fig. 2. *The cost functions for Example 3.2.*

5 Conclusion

Despite all the problems researchers have had with the solutions of non-linear fractional-order initial-valued fractional Volterra-type differential equations, a certain iterative method based on the combination of neural networks and power series has assisted them. It can be included that fractional differential equations can not be typically solved analytically. Several neural network methods are proven to provide solutions to such problems. In fact, here, the principles and background information on ANNs and of some existing numerical methods are the most readily available source of solving problems. The main purpose of this paper is to introduce an iterative method based on the combination of neural networks and power series for solving non-linear fractional Volterra-type differential equations. Considering the mentioned source of obtaining solutions, we have also established some basic concept of fractional calculus which enables us to understand the mentioned fractional problems and to solve them. Contrary to existing methods in the literature, major problems can be dealt with using neural network architecture. Regarding the accuracy and validity characteristics of the proposed method, solutions made by using this method are highly acceptable. Evaluation of the solution is another characteristic which is employed once the network is trained. A new method for construction of a better trail solution for all kinds of fractional initial value problems can be verified and enhanced by combining neural networks to other existing methods by researchers. However, in order to consolidate the validity of such methods, they should be subjected to further investigations by the multiple authorities.

References

[1] S. A. Belbas, Floquet theory for integral and integro-differential equations, *Appl. Math. Comput.* **223**(15), 327-345 (2013).
[2] J. Manafianheris, Solving the integro-differential equations using the modified laplace Adomian decomposition method, *J. Math. Exten.* **6**(1), 41-55 (2012).
[3] S. Yuzbasi, M. Sezer and B. Kemanci, Numerical solutions of integro-differential equations and application of a population model with an improved Legendre method, *Appl. Math. Model.* **37**(4), 2086-2101 (2013).
[4] A. Babaaghaie and K. Maleknejad, Numerical solution of integro-differential equations of high order by wavelet basis, its algorithm and convergence analysis, *J. Comput. Appl. Math.* **325**(1), 125-133 (2017).
[5] A. Bellour and M. Boussel, Numerical solution of delay integro-differential equations by using Taylor collocation method, *Math. Meth. Appl. Sci.* **37**(10), 1491-1506 (2014).
[6] A. Jafarian, P. Ghaderi, A. K. Golmankhaneh and D. Baleanu, Analytical treatment of system of Abel integral equations by homotopy analysis method, *Rom. Rep. Phys.* **66**(3), 603-611 (2014).
[7] A. Jafarian, R. Jafrari and A. K. Golmankhaneh, Solving fully fuzzy polynomials using feed-back neural networks, *Int. J. Comput. Math.* **92**(4),742-755 (2015).
[8] D. Baleanu, A. K. Golmankhaneh and A.K. Golmankhaneh, Solving of the fractional non-linear and linear Schrödinger equations by homotopy perturbation method, *Rom. J. Phys.* **54**, 823-832 (2009).
[9] A. Jafraian, P. Ghaderi, A.K. Golmankhaneh and D. Baleanu, Construction of soliton solution to the Kadomtsev-Petviashvili-II equation using homotopy analysis method, *Rom. Rep. Phys.* **66**(2), 296-306 (2014) .
[10] M. A. F. Araghi and G. K. Gelian, Numerical solution of integro-differential equations based on double exponential transformation in the sinc-collocation method, *Appl. Math. Comput. Intel.* **1**, 48-55 (2012).

[11] M. Tamamgar, The numerical solution of linear Fredholm integro-differential equations via parametric iteration method, *J. Appl. Comput. Math.* **3**(4), 1-4 (2014).
[12] G. Mehdiyeva, V. Ibrahimov and M. Imanova, On one investigation of numerical solution of integro-differential equations of Volterra type. Problems of Cybernetics and Informatics (PCI), 2012 IV International Conference. IEEE, 2012.
[13] A. K. Golmankhaneh, R. Arefi and D. Baleanu, The proposed modified Liu system with fractional order, *Adv. Math. Phys.* **2013**, Article ID 186037, (2013).
[14] D. Baleanu, S. I. Muslim, E. M. Rabei, A. K. Golmankhaneh and A. K Golmankhaneh, On fractional Hamiltonian systems possessing first-class constraints within Caputo derivatives, *Rom. Rep. Phys.* **63**(1), 3–8 (2011).
[15] D. Baleanu, A. K. Golmankhaneh and A. K. Golmankhaneh, The dual action of fractional multi-time Hamilton equations, *Int. J. Theor. Phys.* **48**(9), 2558–2569 (2009).
[16] V. V. Uchaikin, *Fractional derivatives for physicists and engineers Vol. 1 background and theory. Vol 2. application* Springer, Berlin, 2013.
[17] R. Herrmann, *Fractional calculus: an introduction for physicists*, World Scientific, 2014.
[18] X. Ma and Ch. Huang, Numerical solution of fractional integro-differential equations by a hybrid collocation method, *Appl. Math. Comput.* **219**(12), 6750–6760 (2013).
[19] S. Mashayekhi and M. Razzaghi, Numerical solution of nonlinear fractional integro-differential equations by hybrid functions, *Eng. Anal. Bound. Elem.* **56**, 81–89 (2015).
[20] Z. Taheri, S. Javadi and E. Babolian, Numerical solution of stochastic fractional integro-differential equation by the spectral collocation method, *J. Comput. Appl. Math.* **321**, 336–347(2017).
[21] D. S. Mohammed, Numerical solution of fractional integro-differential equations by least squares method and shifted Chebyshev polynomial, *Math. Probl. Eng.* **7**(4), 1589–1596 (2016).
[22] T. Oyedepo, O. Taiwo, J. Abubakar and Z. Ogunwobi, Numerical studies for solving fractional integro-differential equations by using least squares method and Bernstein polynomials, *Fluid Mech.* **3**(142), 2476–2296 (2016).
[23] S. Shahmorad, S. Pashaei and M. S. Hashemi, Numerical solution of a nonlinear fractional integro-differential equation by a geometric approach, *Differ. Equ. Dyn. Syst.* **(2017)**, doi:10.1007/s12591-017-0395-1.
[24] Y. Wang and L. Zhu, Solving nonlinear Volterra integro-differential equations of fractional order by using Euler wavelet method, *Adv. Differ. Equ.* **2017**(1), 27 (2017).
[25] A. Jafarian, M. Mokhtarpour and D. Baleanu, Artificial neural network approach for a class of fractional ordinary differential equation, *Neural Comput. Appl.* **28**(4), 765-773 (2017).
[26] A. Jafarian, F. Rostami, A. K. Golmankhaneh and D.Baleanu, Using ANNs approach for solving fractional order Volterra integro-differential equations, *Int. J. Comput. Int Sys.* **10**(1), 470–480 (2017).
[27] F. Rostami and A. Jafarian, A new artificial neural network structure for solveing high-order linear fractional differential equations, *Int. J. Comput. Math.* **95**(3), 528–539 (2018).
[28] M. A. Z. Raja, S. A. Niazi and S. A. Butt, An intelligent computing technique to analyze the vibrational dynamics of rotating electrical machine, *Neurocomput.* **219**, 280–299 (2017) .
[29] S. Mall and S. Chakraverty, Application of Legendre neural network for solving ordinary differential equations, *Appl. Soft Comput.* **43**, 347–356 (2016).
[30] M. A. Z. Raja, M. A. Manzar and R. Samar, An efficient computational intelligence approach for solving fractional order Riccati equations using ANN and SQP, *Appl. Math. Model.* **39**(10-11), 3075–3093 (2015).
[31] M. A. Z. Raja, S. A. Niazi, S. A. Butt, An intelligent computing technique to analyze the vibrational dynamics of rotating electrical machine, *Neurocomputing,* **219**, 280-299 (2017).
[32] A. D. Dongare, R. R. Kharde and A. D. Kachare, Introduction to artificial neural network, *Int. J. Eng.Inn. Tech.* **2**(1), 189–194 (2012).
[33] K. Patan, *Artificial neural networks for the modelling and fault diagnosis of technical processes*, Springer-Verlag, Berlin, Chennai, 2008.
[34] G. A. Anastassiou, *Intelligent systems: approximation by artificial neural networks,* Springer-Verlag, Berlin, Chennai, 2011.
[35] D. Graupe, *Principles of artificial neural networks* (2nd Edition), World Scientific Publishing, 2007.
[36] A. Jafarian, S. Measoomy Nia, A. K. Golmankhaneh and D. Baleanu,On artificial neural networks approach with new cost functions, *Appl. Math. Comput.* **339**, 546-555 (2018).
[37] A. K. Golmankhaneh and D. Baleanu, Non-local integrals and derivatives on fractal fets with applications, *Open Phys.* **14**(1), 542-548 (2016) .
[38] M. H. Hassoun, *Fundamentals of artificial neural networks,* MIT Press, 1995.