Artificial Neural Network Approach for Solving Fuzzy Fractional order initial value problems under H-differentiability

Somayeh Ezadi[a], Tofigh Allahviranloo[b*]
[a] Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran,
[b] Faculty of engineering and natural sciences, Bahcesehir university, Istanbul, Turkey.

Abstract
This paper aims to solve the celebrated Fuzzy Fractional Differential Equations (FFDE) using an Artificial Neural Network (ANN) technique. Compared to the integer order differential equation, the proposed FFDE can better describe several real application problems of various physical systems. To accomplish the aforementioned aim, the error back propagation algorithm and a multi-layer feed forward neural architecture are utilized using the unsupervised learning in order to minimize the error function as well as the modification of the parameters such as weights and biases. By combining the initial conditions with the ANN, output provides an appropriate approximate solution of the proposed FFDE. Then, two illustrative examples are solved to confirm the applicability of the concept as well as to demonstrate both the precision and effectiveness of the developed method. By comparing with some traditional methods, the obtained results reveals a close match that confirms both accuracy and correctness of the proposed method.
Keywords: Fuzzy Fractional Differential Equations (FFDE); Artificial Neural Network (ANN); Back propagation algorithm; Unsupervised learning.

1. Introduction
In recent decades, many engineering and scientific problems required the fuzzy fractional calculations, as they can model more accurately than the other expected systems. The real-world applications of FFDEs shortly have attracted a considerable attention in mathematics and in the scientific societies [1-4,8,22-25]. For instance, fuzzy fractional calculations have been developed to model nonlinear fluctuations in earthquakes, fluid dynamics, frequency dependent sampling behavior, many viscoelastic materials, statistical and continuum machines, colored Noise, solid mechanics, economics, signal processing the theory of control and dynamics of interactions between nanoparticles and layers. The reason for this is that it is well known as a realistic model of a physical phenomenon, which contains a dependency on a time constant and also a previous time history that can be successfully obtained by using fractional calculations. Mathematical formulas of many of the phenomena mentioned, contain differential equations with a degree of fractional or fuzzy fraction. The theory of fuzzy sets is a powerful method for modeling vague processes and mathematical information [27]. Here, it is worthwhile to mention that Chang and Zadeh [7] first developed the fuzzy concept, stating as the others jumped Dubois extended issue [9]. In the following, the Fuzzy Differential Equations (FDE) along with the initial value problem was demonstrated by Kalva and Seikkala [14].

In last few years, on the other hand, various machine intelligence procedures in particular Artificial Neural Network (ANN) methods have been established as a powerful technique to solve a variety of real-world problems because of its excellent learning capacity [11, 28]. ANN approach has attracted much consideration to its advantages such as learning, adaptive, error computation, fault-tolerance and, etc. Recently, many attentions have been paid to utilize ANN to solve ordinary [6,16], differential equations of fractional order [18] and fuzzy differential equations [10]. Qu and Liu [19] described the cosine basis functions as well as the neural network training along with adjustable parameters to solve single and the

Correspondence: Tofigh Allahviranloo, **Email: tofigh.allahviranloo@eng.bau.edu.tr**

systems of the coupled fractional order differential equations. Rostami and Jafarian [20] employed the combination of power series method and the ANN approach to handle higher order linear fractional differential equations. Sabouri et al. [21] employed the ability of neural networks for solving fractional order optimal control problems. In another study [13] Jafarian et al. have used combination of multi-layer ANN and the power series method to solve a class of fractional order initial value problems. In this investigation, the authors vested their effort to develop a multi-layer ANN model with unsupervised back propagation learning algorithm to solve the fuzzy fractional order initial value problems. In this way, the ANN approximate solution of FFDE is provided as the sum of two terms as follows: the first one meets the initial/boundary conditions, whereas the second part contains output of neural network including the adjustable parameters. In the following, the error back propagation principles (gradient descent procedure) and feed forward neural network model are implemented to modify the network parameters as well as to minimize the computed error function. After that, the obtained initial weights from input to hidden and then from hidden to output layer, are randomly determined. The approximate solution of the proposed FFDE using the neural network is achieved to be very practical, even though it depends on the ANN model concerning the considered individual. The structure of the paper encompasses the following sections:
In section two, the basic concepts are expressed. In section 3, the algorithm of proposed method for approximate solution of fuzzy fractional order initial value problem and error estimation are described. In section 4, numerical examples are presented, in section 5, the applied examples are presented, and finally, in section 6, the conclusion is expressed.

2. Preliminaries

In this section, some definitions are brought. Also, we suppose that the H-difference exist throughout the paper.

Denote $R_F=\{u:R^n\to[0,1]\vee u\,satisfies\,(i)-(iv)\,below\}$, where

(i) u is normal, i.e. there exists $x\in R^n$, such that $u(x)=1$.

(ii) u is fuzzy convex, i.e.

$$\forall x, y\in R^n\wedge\lambda\in[0,1], u(\lambda x+(1-\lambda)y)\geq min\{u(x),u(y)\}.$$

(iii) u is upper semi-continuous.

(iv) $cl\{s\in R^n\vee u(s)>0\}$, is compact in $x\in R^n$.

Then R_F is called the space of fuzzy numbers.

The α-level set of a fuzzy number $u\in R_F, 0\leq\alpha\leq 1$, denoted $[u]_\alpha=\{x\in R^n\vee u(x)>\alpha=[\underline{u}(\alpha),\bar{u}(\alpha)]$. Then from (i) to (iv), it follows that the α-level set $[u]_\alpha$ is a closed interval for all $\alpha\in[0,1]$. A triangular fuzzy number is defined as a fuzzy set in R_F, that is specified by an ordered triple $u(a,b,c)\in R^3$with $a\leq b\leq c$ such that $\underline{u}(\alpha)=a+(b-a)\alpha$ and $\bar{u}(\alpha)=c-(c-b)\alpha$ are the endpoints of α -level sets for all $\alpha\in[0,1]$.

Definition 2.1. Let $f:(a,b)\to R_F$ is Hukuhara differentiable at $x_0\in(a,b)\subseteq R$ if for some$h_0>0$ the Hukuhara difference $f(x_0+\Delta x)-¿_h f(x_0), f(x_0)-¿_h f(x_0+\Delta x), ¿¿$ exist in R_F for all $0<\Delta<h_0$ and if there exist an element $f'(x_0)\in R_F$ such that

$$\lim_{\Delta x\to 0^{+¿}} d_\infty ¿¿$$

and

$$\lim_{\Delta x\to 0^{+¿}} d_\infty ¿¿$$

The fuzzy number valued function $f'(x_0)$ is called the Hukuhara derivative of f at x_0, [5]. Recall that $U-¿_h V=W\in R_F¿$ are defined on level sets, where $[U]_\alpha -¿_h[V]_\alpha=[W]_\alpha ¿$ for all $0\le\alpha\le 1$. By consideration of definition of the metric d_∞, all the level set mappings $[f(.)]_\alpha$ are Hukuhara differentiable at x_0 with Hukuhara derivatives $[f'(x_0)]_\alpha$ for each $0\le\alpha\le 1$, when $f:(a,b)\to R_F$ is Hukuhara differentiable at $x_0$with Hukuhara derivative $f'(x_0)$.

Theorem 2.2. Let $f(x)$ be a fuzzy-valued function on $(-\infty,\infty)$ and it is represented by $f(x;\alpha)=[\underline{f}(x;\alpha),\overline{f}(x;\alpha)]$ for any fixed $\alpha\in[0,1]$. Assume that $¿\underline{f}(x;\alpha)\vee ¿$ and $¿\overline{f}(x;\alpha)\vee ¿$ are Riemann integrable on $(-\infty,\infty)$ for all $\alpha\in[0,1]$. Then $f(x)$ is improper fuzzy Riemann-integrable on $(-\infty,\infty)$ and the improper fuzzy Riemann integral is a fuzzy number. Furthermore, we have [26].

Definition 2.3. Suppose that $f:V\subseteq R\to R_F$ is fuzzy-valued function with $[f(x)]_\alpha=[\underline{f}(x;\alpha),\overline{f}(x;\alpha)]$. If the partial derivatives of $\underline{f}(x;\alpha)$ and$\overline{f}(x;\alpha)$ with respect to $x\in R$ exist and the interval $[\underline{f}'(x;\alpha),\overline{f}'(x;\alpha)]$ defines the α-level set of a fuzzy number for $x\in R$, $\alpha\in[0,1]$. Then For $x\in R,\alpha\in[0,1]$, $f(x)$ is called differentiable and we write,
$f'(x;\alpha)=(\underline{f}'(x;\alpha),\overline{f}'(x;\alpha)).(2.3)$

Definition 2.4. (See [18].) Let $f:[0,b]\subseteq R\to R$be continuous and Lebesque integrable real-valued function. The modified Riemann-Liouville fractional derivative of function $f(x)$ of order $,0<¿1$, is denoted by ${}^{mRL}D^\beta f(x)$ and defined as:

$$ {}^{mRL}D^\beta f(x)=\frac{1}{\Gamma(1-\beta)}\frac{d}{dx}\int_0^x\frac{f(t)-f(0)}{(x-t)^\beta}dt,0<\beta<1,x>0(2.4)$$

The definition of modified Riemann-Liouville fuzzy fractional derivative based on an integration of Definition 2.4, H- difference, and H- derivative is introduced below.

Definition 2.5. Let$f:[a,b]\subseteq R\to R_F$ be continuous and Lebesque integrable fuzzy-valued function,
$G(x)=\frac{1}{\Gamma(1-\beta)}\int_a^x f(t)\frac{-¿_h f(a)}{(x-t)^\beta}dt.¿$ The function $f(x)$ is modified Riemann-Liouville fuzzy fractional differentiable of order $0<¿1$at $x_0\in(a,b)$ if there exists an element ${}^{mRL}D^\beta f(x_0)\in R_F$ such that for $h>0$ sufficiently near zero the following limit exists

$$\lim_{h\to 0}G(x_0)\frac{-¿_h G(x_0+h)}{-h}=G(x_0-h)\frac{-¿_h G(x_0)}{-h}={}^{mRL}D^\beta f(x_0)(2.5)¿¿$$

Or

$$\lim_{h\to 0}G(x_0+h)\frac{-¿_h G(x_0)}{h}=G(x_0)\frac{-¿_h G(x_0-h)}{h}={}^{mRL}D^\beta f(x_0)(2.6)¿¿$$

2.6. Multi-layer perceptron (MLP)
A multi-layer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that utilizes a nonlinear activation function. MLP uses a supervised learning technique so-called backpropagation for training. Its

multiple layers and non-linear activation distinguish MLP from a linear perceptron. Multi-layer networks use a variety of learning techniques, in which the most popular being back-propagation (BP), is based on the error correction learning rule. Therefore, to calculate the sensitivities for the different layers of neurons in the MLP network, the derivative of conversion neurons functions is required. To do so, those functions should be used that have derivative. One of these functions is the Sigmoid function defined as follows $\sigma(n)=\frac{1}{1+e^{-n}}$. The output of this function values in the range [0, 1]. Moreover, it can be stated very well and very badly, as shown in Figure 1.

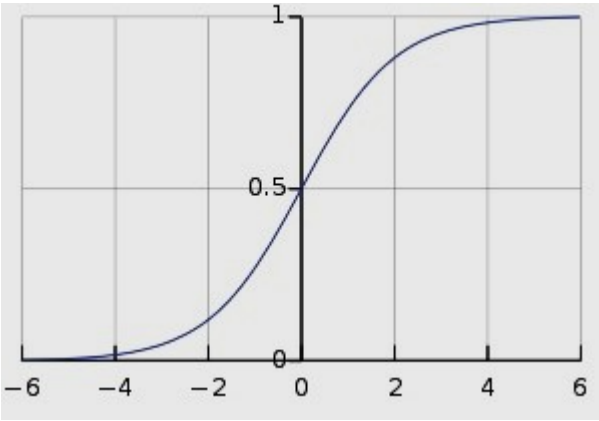

Figure 1. shows a hyperbolic tangent function.

Theorem 2.7. (The World Approximation Builder)
The MLP network with one hidden layer with a sigmoid function (Hyperbolic tangent function) in the middle layer and linear transformation functions in output layer are able to approximate all functions in any degree of the integral of the square (see [12]).

2.8. Broyden Fletcher Goldfarb Shanno Teqnique (BFGs Teqnique)
To compute the value of some predefined error-function, the obtained output values are compared with the exact solution. By various techniques, the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state where the error of the calculations is small. In this case, one would say that the network has learned a certain target function. To adjust weights properly, we should minimize the unconstrained optimization problem. To achieve this aim, some minimization techniques such as the steepest descent method and the conjugate gradient or Quasi-Newton methods can be employed. The Newton method is one of the important algorithms in nonlinear optimization. The main disadvantage of the Newton method is that it is necessary to evaluate the second derivative matrix (Hessian matrix). Quasi-Newton methods were originally proposed by Davidon in 1959 and were later developed by Fletcher and Powell (1963). The most fundamental idea in Quasi-Newton methods is the requirement to calculate an approximation of the Hessian matrix. Here the Quasi-Newton BFGS method is used (see [15]).

3. Algorithm of approximate solution of fuzzy fractional order initial value problem and error estimation

In this section, the general formulation of ANN model is discussed for the proposed fuzzy fractional order differential equation. In particular, structure of ANN model and the formulations for initial value fractional order fuzzy problems are incorporated in detail.

3.1. ANN formulation for fuzzy FFDE

Suppose a fuzzy fractional order initial value problem as follows:

$$\begin{cases} {}_{x_0}D_x^{\beta}\tilde{y}(x)=f\left(x,\tilde{y}(x)\right), 0<\beta\le 1, \\ \text{¿}\ \tilde{y}(x_0)=\tilde{y}_0\in R_F \end{cases} \quad (3.1)$$

Suppose $\tilde{y}_N(x,p)$ denotes the approximate solution of ANN model with p which is a vector containing corresponding weights and x is the input data. The above FFDE is transformed into the following problem

$${}_{x_0}D_x^{\beta}\tilde{y}_N(x,p)=f\left(x,\tilde{y}_N(x,p)\right) \quad (3.2)$$

The approximate solution $\tilde{y}_N(x,p)$ of feed forward neural network using the network parameters p may be described in the following form

$$\tilde{y}_N(x,p)=\tilde{A}(x)+F\left(x,\widetilde{Net}(x,p)\right) \quad (3.3)$$

Here, it should be noted that the first term $\tilde{A}(x)$ in right hand side does not contain adjustable parameters and satisfies only initial\boundary conditions, whereas the second term $F\left(x,\widetilde{Net}(x,p)\right)$ contain the single output $\widetilde{Net}(x,p)$of feed forward neural network using input x and vector containing the corresponding weights p. Here we propose a three layer ANN model with one input node x, one hidden layer consisting of k number of nodes and one output node $\widetilde{Net}(x,p)$. The output $\tilde{N}(x,p)$ can be expressed as follows:

$$\widetilde{Net}(x,p)=\sum_{j=1}^{K}\tilde{v}_j\sigma(\tilde{z}_j), \tilde{z}_j=\tilde{w}_j x+\tilde{b}_j, K\in N \quad (3.4)$$

where $\tilde{w}_j$ denotes weight from input to j^{th} hidden unit,$\tilde{v}_j$ means the weight from j^{th} hidden unit to output unit and finally $\tilde{b}_j$ refers to the bias for j^{th} hidden node. The Sigmoid function, (σ), is already described in Section 2.6. The General form of corresponding error function for the fractional order initial value problem may be created as

$$E(x,p)=\sum_{i=1}^{h}\left({}_{x_0}D_x^{\beta}\tilde{y}_N(x_i,p)-f\left(x_i,\tilde{y}_N(x,p)\right)\right)^2, h\in N \quad (3.5)$$

3.2. Error back propagation learning algorithm

The Error back propagation learning algorithm is employed to update the network parameters (weights) and for minimizing error function of the ANN. For FFDE we consider an unsupervised version of back propagation method. Here gradient descent method has been used for updating the parameters.

$$\tilde{w}_j^{k+1}=\tilde{w}_j^{k}+\Delta\tilde{w}_j^{k}=\tilde{w}_j^{k}+\left(-\eta\frac{\partial\tilde{E}(x,\Omega)^{k}}{\partial\tilde{w}_j^{k}}\right) \quad (3.6)$$

$$\tilde{v}_j^{k+1}=\tilde{v}_j^k+\Delta\tilde{v}_j^k=\tilde{v}_j^k+\left(-\eta\frac{\partial\tilde{E}(x,\Omega)^k}{\partial\tilde{v}_j^k}\right)(3.7)$$

where is learning parameter, m is iteration step which is used to update the weights as usual in ANN and $\tilde{E}(x,p)$ is the error function. Here, k is the number of iterations

3.3 Formulation of fuzzy fractional order initial value problem for β ¿

Let us consider a FFDE of order β ¿

$$\begin{cases} {}_{x_0}D_x^{\beta}\tilde{y}(x)=f(x,\tilde{y}(x)), \\ \text{¿}\,\tilde{y}(x_0)=\tilde{y}_0\in R_F \end{cases}(3.8)$$

The approximate solution of ANN may be written as

$$\tilde{y}_N(x,p)=\tilde{y}_0+(x-x_0)\widetilde{Net}(x,p)(3.9)$$

When $x=x_0$, obtained approximate solution $\tilde{y}_N(x,p)$ can satisfy the initial condition. The error function is computed using relation (3.5).

3.4. Structure of multi-layer ANN model for FDE

In this subsection, a three layers ANN model is proposed for the problem at hand. Figure 2 illustrates the structure of neural network architecture including an input layer with single input node and a bias, one hidden layer having five hidden nodes and output layer consisting of one output node. In this network, the initial weights of $\tilde{w}_j$from input to hidden layer as well as $\tilde{v}_j$ from hidden to output layer, are randomly determined.

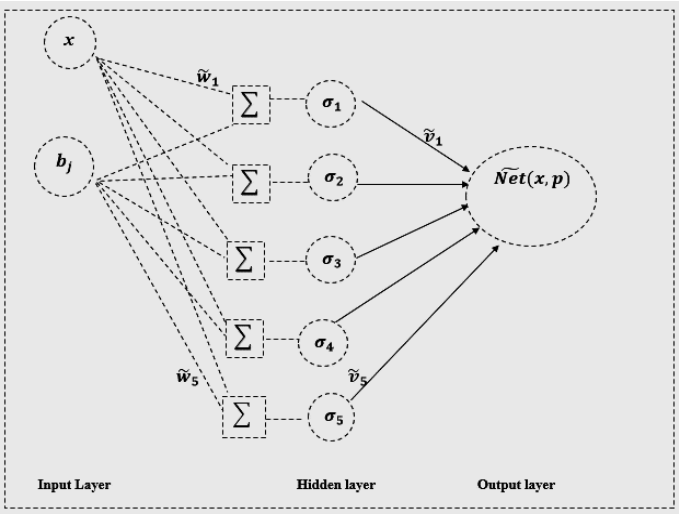

Figure 2. Proposed Artificial neural network architecture

3.5 Computation of gradient for Fractional order initial value problem

In this section, we differentiate $\widetilde{E}(x,p)$ in order to minimize the error function $\widetilde{E}(x,p)$ that is to update the network parameters (weights). As a result, regarding to their inputs, the gradient of network output is computed as follows.

With the $\alpha-$¿level set, the fractional derivatives (according to conformable fractional derivative) of $\widetilde{Net}(x,p)$ with respect to input x is written as

$$D_x^{\beta}\widetilde{Net}(x,p)=\left[min\left(D_x^{\beta}\underline{Net}(x,p),D_x^{\beta}\overline{Net}(x,p)\right),max\left(D_x^{\beta}\underline{Net}(x,p),D_x^{\beta}\overline{Net}(x,p)\right)\right] \quad (3.10)$$

Where

$D_x^{\beta}\underline{Net}(x,p)=\underline{v}_j\sigma'(\underline{z}_j)\underline{w}_j x^{1-\beta}$,

¿ $D_x^{\beta}\overline{Net}(x,p)=\overline{v}_j\sigma'(\overline{z}_j)\overline{w}_j x^{1-\beta}$.

Let $\left(D_x^{\beta}\widetilde{Net}(x,p)\right)=\widetilde{Net}_r$ denotes the derivative of the network output with respect to its inputs. regarding the other parameters, the derivative of $\widetilde{Net}_r$ may be obtained as follows (based on the conformable fractional derivative rules):

$$(1)\frac{\partial\widetilde{Net}_r}{\partial\widetilde{w}_j}=\left[min\left(\frac{\partial\underline{Net}_r}{\partial\underline{w}_j},\frac{\partial\overline{Net}_r}{\partial\overline{w}_j}\right),max\left(\frac{\partial\underline{Net}_r}{\partial\underline{w}_j},\frac{\partial\overline{Net}_r}{\partial\overline{w}_j}\right)\right],(3.11)$$

where

$\frac{\partial\underline{Net}_r}{\partial\underline{w}_j}=\underline{v}_j x^{1-\beta}\left(\sigma'(\underline{z}_j)+\sigma''(\underline{z}_j)\underline{w}_j x\right)$,

¿¿ $\frac{\partial\overline{Net}_r}{\partial\overline{w}_j}=\overline{v}_j x^{1-\beta}\left(\sigma'(\overline{z}_j)+\sigma''(\overline{z}_j)\overline{w}_j x\right)$.

$$(2)\frac{\partial\widetilde{Net}_r}{\partial\widetilde{v}_j}=\left[min\left(\frac{\partial\underline{Net}_r}{\partial\underline{v}_j},\frac{\partial\overline{Net}_r}{\partial\overline{v}_j}\right),max\left(\frac{\partial\underline{Net}_r}{\partial\underline{v}_j},\frac{\partial\overline{Net}_r}{\partial\overline{v}_j}\right)\right],(3.12)$$

where

$\frac{\partial\underline{Net}_r}{\partial\underline{v}_j}=\underline{w}_j x^{1-\beta}\sigma'(\underline{z}_j)$,

¿¿ $\frac{\partial\overline{Net}_r}{\partial\overline{v}_j}=\overline{w}_j x^{1-\beta}\sigma'(\overline{z}_j)$.

$$(3)\frac{\partial\widetilde{Net}_r}{\partial\widetilde{b}_j}=\left[min\left(\frac{\partial\underline{Net}_r}{\partial\underline{b}_j},\frac{\partial\overline{Net}_r}{\partial\overline{b}_j}\right),max\left(\frac{\partial\underline{Net}_r}{\partial\underline{b}_j},\frac{\partial\overline{Net}_r}{\partial\overline{b}_j}\right)\right],(3.13)$$

where

$$\frac{\partial \underline{Net}_r}{\partial \underline{b}_j} = \underline{w}_j \underline{v}_j x^{1-\beta} \sigma'\left(\underline{z}_j\right)$$

¿¿ $\frac{\partial \overline{Net}_r}{\partial \overline{b}_j} = \overline{w}_j \overline{v}_j x^{1-\beta} \sigma'\left(\overline{z}_j\right)$

Here

$$Net(x,p) = \sum_{j=1}^{m} v_j \sigma(z_j), z_j = w_j x + b_j$$

From relation (3.9) we have (by differentiating)

$$ {}_{x_0}D_x^{\beta} \tilde{y}_N(x) = \left[min\left(D_x^{\beta} \underline{y}_n(x,p), D_x^{\beta} \overline{y}_n(x,p) \right), max\left(D_x^{\beta} \underline{y}_n(x,p), D_x^{\beta} \overline{y}_n(x,p) \right) \right], \quad (3.14)$$

where

$$ {}_{x_0}D_x^{\beta} \underline{y}_n(x) = (x-x_0)^{1-\beta} \left(\underline{Net}(x,p) + (x-x_0) \left(D_x^{\beta} \underline{Net}(x,p) \right) \right)$$

¿¿ $ {}_{x_0}D_x^{\beta} \overline{y}_n(x) = (x-x_0)^{1-\beta} \left(\overline{Net}(x,p) + (x-x_0) \left(D_x^{\beta} \overline{Net}(x,p) \right) \right)$

After simplifying the above equation, we have

$$ {}_{x_0}D_x^{\beta} \underline{y}_n(x) = (x-x_0)^{1-\beta} \left(\underline{Net}(x,p) + (x-x_0) \left(\underline{w}_j \underline{v}_j x^{1-\beta} \varphi'\left(\underline{z}_j\right) \right) \right)$$

¿¿ $ {}_{x_0}D_x^{\beta} \overline{y}_n(x) = (x-x_0)^{1-\beta} \left(\overline{Net}(x,p) + (x-x_0) \left(\overline{w_j} \overline{v}_j x^{1-\beta} \varphi'\left(\overline{z}_j\right) \right) \right)$

4. Numerical Examples

To better understand the proposed method, these following informative example problems are included. Afterwards, the approximate results by ANN model are compared with analytical\existing numerical solution of each example in order to demonetarize the powerfulness of the proposed method.

Example 4.1. Let us consider a fractional fuzzy differential equation

¿

According to $D_2 \tilde{y}(x)(x)$using the ANN method, the obtained approximate solution of the considered problem via $\alpha = 0, 0.2, 0.4, 0.5, 0.6, 0.8, 1$ is reported in Tables 4.1. In addition, Figure 3. shows the approximate answer.

Table 4.1. Approximation of $\underline{y}^{\alpha}$ and $\overline{y}^{\alpha}$ for Example 4.1.

	$\underline{y}^{\alpha}$							$\overline{y}^{\alpha}$						
x / α	**0**	**0.2**	**0.4**	**0.5**	**0.6**	**0.8**	**1**	**0**	**0.2**	**0.4**	**0.5**	**0.6**	**0.8**	**1**
0	0.5	0.6	0.7	0.75	0.8	0.9	1	1.5	1.4	1.3	1.25	1.2	1.1	1
0.1	0.414	0.496	0.579	0.620	0.662	0.745	0.827	1.241	1.159	1.076	1.035	0.993	0.910	0.827
0.2	0.355	0.426	0.497	0.533	0.569	0.640	0.711	1.067	0.995	0.924	0.889	0.857	0.782	0.711
0.3	0.316	0.379	0.442	0.474	0.505	0.568	0.632	0.948	0.884	0.821	0.790	0.758	0.695	0.632
0.4	0.285	0.343	0.400	0.428	0.457	0.514	0.571	0.857	0.800	0.743	0.714	0.686	0.628	0.571
0.5	0.261	0.313	0.365	0.392	0.418	0.470	0.522	0.783	0.731	0.679	0.653	0.627	0.575	0.522
0.6	0.240	0.289	0.337	0.361	0.385	0.433	0.481	0.722	0.674	0.626	0.602	0.578	0.529	0.481
0.7	0.223	0.268	0.312	0.335	0.357	0.402	0.446	0.670	0.625	0.580	0.558	0.536	0.491	0.446
0.8	0.208	0.249	0.291	0.312	0.333	0.374	0.416	0.624	0.582	0.541	0.521	0.499	0.458	0.416
0.9	0.194	0.233	0.272	0.292	0.311	0.350	0.389	0.584	0.545	0.506	0.487	0.468	0.429	0.389
1	0.183	0.219	0.256	0.274	0.293	0.329	0.366	0.549	0.513	0.476	0.458	0.439	0.403	0.366

$\underline{y}^{\alpha}, \overline{y}^{\alpha}$ x

Figure 3. The approximate answer of Example 4.1.

Example 4.2. Let us consider a fractional fuzzy differential equation

¿

According to ${}^{c}D_{1}[y]^{z}(x)$ using the ANN method, the obtained approximate solution of the considered problem via $\alpha = 0, 0.2, 0.4, 0.5, 0.6, 0.8, 1$ is reported in Tables 4.2. In addition, Figure 4. shows the approximate answer.

Table 4.2. Approximation of $\underline{y}^{\alpha}$ and $\overline{y}^{\alpha}$ for Example 4.2.

	$\underline{y}^{\alpha}$						$\overline{y}^{\alpha}$					
x / α	**0**	**0.2**	**0.4**	**0.6**	**0.8**	**1**	**0**	**0.2**	**0.4**	**0.6**	**0.8**	**1**
0	0.5	0.6	0.7	0.8	0.9	1	1.5	1.35	1.25	1.15	1.05	1
0.1	1.5396	1.7462	1.9472	2.1437	2.3367	2.5267	3.4444	3.1737	2.9913	2.8070	2.6207	2.5267
0.2	3.1421	3.4657	3.7771	4.0790	4.3732	4.6609	6.0299	5.6291	5.3577	5.0824	4.8026	4.6609
0.3	5.1875	5.6298	6.0536	6.4624	6.8594	7.2464	9.0740	8.5411	8.1793	7.8114	7.4367	7.2464
0.4	7.6859	8.2537	8.7961	9.3185	9.8245	10.3169	12.6311	11.9580	11.5003	11.0342	10.5586	10.3169
0.5	10.6872	11.3919	12.0639	12.7101	13.3352	13.9426	16.7883	15.9622	15.3996	14.8262	14.2406	13.9426
0.6	14.2580	15.1145	15.9304	16.7139	17.4711	18.2062	21.6417	20.6456	19.9667	19.2743	18.5665	18.2062
0.7	18.4758	19.5023	20.4791	21.4164	22.3214	23.1994	27.2943	26.1083	25.2994	24.4738	23.6294	23.1994
0.8	23.4285	24.6462	25.8038	26.9139	27.9850	29.0234	33.8590	32.4597	31.5048	30.5297	29.5319	29.0234
0.9	29.2152	30.6481	32.0096	33.3142	34.5724	35.7914	41.4605	39.8212	38.7020	37.5586	36.3881	35.7914
1	35.9470	37.6228	39.2140	40.7381	42.2070	43.6297	50.2977	48.3282	47.0239	45.6910	44.3258	43.6297

$\underline{y}^{\alpha}, \overline{y}^{\alpha}$ x

Figure 4. The approximate answer of Example 4.2.

5. Conclusions

This paper presented a new approach to solve fractional order fuzzy differential equations by using artificial neural network model. In the following, the accuracy of the proposed method has been investigated by solving several FDEs. Besides, as the algorithm has been unsupervised, the error back propagation algorithm was exploited to minimize the error function. In this paper, the corresponding initial weights from input to hidden as well as from hidden to output, has been randomly specified. The obtained approximate solution was differentiable and closed. According to the tables and Figs., the approximate solution by ANN reveals that the results are accurately achieved. Ultimately, it can be concluded that the proposed ANN algorithm is computationally efficient, simple, and straight forward.

References

[1] S. Ahmad, A. Ullah, K. Shah, S. Salahshour, A. Ahmadian, and T. Ciano, Fuzzy fractional-order model of the novel coronavirus, Advances in Difference Equations, (2020) 1-17.

[2] T.M. Atanackovic, S. Pilipovic, B. Stankovic, D. Zorica, Fractional Calculus with Applications in Mechanics: From the Cell to the Ecosystem, Wiley-ISTE (2014).

[3] R. L. Bagley, On the fractional order initial value problem and its engineering applications, in: Fractional Calculus and Its Applications (Ed. K. Nishimoto), Tokyo, College of Engineering, Nihon University, (1990) 12-20.

[4] D. Baleanu , K. Diethelm , E. Scalas , J.J. Trujillo , Fractional calculus models and numerical methods, Series on Complexity, Nonlinearity and Chaos, World Scientific, (2012).

[5] B. Bede, and Gal. S. G., Generaizations of the differentiability of fuzzy-number-valued functions with applications to fuzzy differential equations, Fuzzy Sets and Systems, 151 (2005) 581–599.

[6] S. Chakraverty, Susmita Mall, Artificial Neural Networks for Engineers and Scientists: Solving Ordinary Differential Equations', CRC Press/Taylor & Francis Group (2017).

[7] Chang, S.S.L. and L. Zadeh, On fuzzy mapping and control, IEEE Transactions on System, Man and Cybernetics 2 (1972) 30-34.

[8] K. Diethelm, N. J. Ford, Analysis of fractional differential equations, J. Math. Anal. Appl, 265 (2002) 229-248.

[9] D. Dubois, and H. Prade, Towards fuzzy differential calculus, Fuzzy Sets and Systems 8 (1982) 225-233.

[10] S. Ezadi, N. Parandin, A. GHomashi, Numerical Solution of Fuzzy Differential Equations Based on Semi-Taylor by Using Neural Network, J. Basic. Appl. Sci. Res., (2013) 3(1s) 477-482.

[11] D. Graupe, Principle of artificial neural networks (2nd Edition), World Scientific publishing, (2007).

[12] K. Hornick, M. Stinchcombe, White H. Multilayer feedforward networks are universal approximators, Neural Networks, 2 (1989) 359-366.

[13] A. Jafarian, M. Mokhtarpour, D. Baleanu, Artificial neural network approach for a class of fractional ordinary differential equation, Neural Computation and application 28 (2017) 765-773.

[14] Kaleva,O., Fuzzy differential equations, Fuzzy Sets and Systems 24 (1987) 301-317.

[15] C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. Mathematical Programming, 45(3) (1989) 503-528.

[16] S. Mall and S. Chakraverty, Application of Legendre neural network for solving ordinary differential equations, Applied Soft Computing, 43 (2016) 347-356.

[17] K.S. Miller, B. Ross, An Introduction to the Fractional Calculus and Fractional Differential Equations, Wiley-Interscience Publication, JohnWiley & Sons, Inc. New York, USA, (1993).

[18] M. Pakdaman, A. Ahmadian, S. Effati, S. Salahshour, D. Baleanu., Solving differential equations of fractional order using an optimization technique based on training artificial neural network, Applied Mathematics and Computation 293 (2017) 81–95.
[19] H. Qu, X. Liu, A Numerical Method for Solving Fractional Differential Equations by Using Neural Network, Advances in Mathematical Physics, 2015 (2015) 1-12.
[20] F. Rostami and A. Jafarian., A new artificial neural network structure for solving high-order linear fractional differential equations, international journal of computer mathematics, 95(3) (2017) 528-539.
[21] J. K. Sabouri , S. Effati, M. Pakdaman, A Neural Network Approach for Solving a Class of Fractional Optimal Control Problems, Neural Process Lett 45 (2017) 59–74.
[22] J. Sabatier, M. Aoun, A. Oustaloup, G. Grgoire, F. Ragot, P. Roy, Fractional system identification for lead acid battery state of charge estimation, Signal Process 86 (2006) 2645–2657.
[23] S. Salahshour, T. Allahviranloo, S. Abbasbandy., Solving fuzzy fractional differential equations by fuzzy Laplace transforms, Communications in Nonlinear Science and Numerical Simulation. 17 (3), (2012) 1372-1381.
[24] S. Salahshour, T. Allahviranloo, S. Abbas-bandy, D. Baleanu, Existence and unique-ness result for fractional di eential equa-tions with uncertainty, Advances in Di er-ence Equations22 (2012) 1-12.
[25] V. Uchaikin, Fractional derivatives for physicists and engineers, Springer, Berlin (2013).
[26] H.-C. Wu, The improper fuzzy Riemann integral and its numerical integration, Inf. Sci. 111 (14) (1998) 109–137.
[27] L.A. Zadeh, Fuzzy sets, Inform. and Control., 8 (1965) 338–353.
[28] J.M. Zurada, Introduction to Artificial Neural Network. West Publ. Co (1994).