Journal of Al-Qadisiyah for Computer Science and Mathematics Vol. 15(1) 2023 , pp Math. 14–28

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Numerical Method For Solving Fuzzy Singular Perturbation Problems With Initial Condition

Tabark Aqeel Al-Janabi[a] , Khalid Mindeel Mohammed Al-Abrahemee[b]

[a]Department of mathematics, College of Education, University of Al-Qadisiyah, Iraq. Email: tabark.aqeel98@gmail.com

[b]Department of mathematics, College of Education, University of Al-Qadisiyah,Iraq. Email: khalid.mohammed@qu.edu.iq

ARTICLE INFO

Article history:
Received: 20 /01/2022
Revised form: 17 /02/2023
Accepted: 21 /02/2023
Available online: 24 /02/2023

Keywords:

Fuzzy singular perturbation problem.
Fuzzy artificial neural networks.
Trial Solution.
Minimized Error Function.
Hyperbolic Tangent Activation Function.

ABSTRACT

In this paper, We present a modified approach that makes use of the neuro-fuzzy system to solve fuzzy singular perturbation problems for ODEs with IC. The name of this modified approach is the modified neuro-fuzzy system method (MNFS). The foundation of this novel approach is to swap off each x in the input vector training. set $\vec{x} = (x_1, x_2, \dots, x_n)$, $x_j \in [a, b]$ a first-order polynomial which will be as $\xi(x) = \frac{\lambda}{2}(\vec{x} + 1)$, $\lambda \in (a, b)$. By using MNFS, it is possible to train the neural network outside of the initial and last point range by choosing training points based on the open interval (a, b). By resolving a few numerical cases and comparing the results to those calculated using different numerical techniques, we demonstrate this improved a technique and how neural networks demonstrate yield answers with accurate and strong generalization. The suggested approach is illustrated with a number of instances.

MSC..

https://doi.org/10.29304/jqcm.2023.15.1.1148

1. Introduction

In the numerical analysis of fuzzy singularly perturbed differential equations, numerous fresh concepts emerged. We attempt to highlight the most significant recent changes in our survey, but the selection is unavoidably subjective and reflects our own primary interests. In the previous year, there was a lot of interest in ODEs's numerical solution. The solution of FSPPs has piqued people's interest This kind of issue has appeared in a number involves convection diffusion processes, optimal control, chemical reactor theory, applied mathematics in the areas of physics, chemistry, mechanics, and fluid dynamics [1]. FSPPs are based on a small , positive parameter that gives the solution a multi - scale character , i.e. in some parts of the region the tris solution changes quickly, whereas in others, it changes more slowly [2]. Recently, parallel processors and artificial neural networks have studied this specific differential equation. Over the past ten years, applications of artificial intelligence (AI) have been more and more widespread, and as a result, a lot of pertinent research has been done. Fuzzy logic, neural networks, genetic

Corresponding author Tabark Aqeel Al-Janabi

Email addresses: tabark.aqeel98@gmail.com

Communicated by 'sub etitor'

programming, and hybrid techniques like neuro fuzzy systems, genetic fuzzy systems, and neural networks based on genetic programming are the main components of AI methodologies. In the realm of artificial intelligence, Fuzzy logic and artificial neural networks are combined in neuro-fuzzy systems, as proposed by Jang [3] in 1993. The fundamental idea behind this (NFS) is that it blends the learning and connectionist structure of neural networks with the human-like reasoning style of fuzzy systems. (NFS) offers strong and adaptable universal approximations with the capacity to investigate IF-THEN rules that are comprehensible. NFS usage is expanding into a variety of fields in our social and technological life. we introduce a modified method for solving fuzzy singular perturbation problems for ODEs. For the purpose of resolving fuzzy singular perturbation issues, this updated approach is known as the modified neuro-fuzzy system (MNFS). This research focuses on developing a novel method to estimate a solution using fuzzy neural networks. To arrive at an approximation of a solution Equations using fuzzy singular differentials. In order to handle the matched uncertainties, When hybrid neuro-fuzzy network systems are applied, the assumption made in some earlier results regarding the uncertainty function regarding border and structural information is removed, which also lowers the conservatism of the acquired results[4]. This structure of neuro – fuzzy system (NFS) can figure out the equivalent output input vectors. In the selection points, the error function is minimal[5,6]. Thus , The suggested MNFS is based on substituting a first-degree polynomial for each element in the training set. We solve numerical instances to illustrate this enhanced approach, we show how this improved method works and compare our findings from other numerical methodologies.

2. Basic definitions

This section presents Some fundamental ideas of fuzzy set theory that are essential to comprehending this topic.

Definition (2.1) [7]: If $\widetilde{U}$is a group of items denoted by the general symbol $\mathfrak{u}$, then a fuzzy set $\mathbb{A}$ in $\widetilde{U}$ is a group of ordered pairs:$\mathbb{A} = \left\{(\mathfrak{u}, \pi_{\mathbb{A}}(\mathfrak{u})) : \mathfrak{u} \in \widetilde{U}, 0 \le \pi_{\mathbb{A}}(\mathfrak{u}) \le 1\right\}$

Where $\pi_{\mathbb{A}}(\mathfrak{u})$ is called the membership function or grade of membership (also degree of compatibility or degree of truth) of $\mathfrak{u}$ in $\mathbb{A}$ that maps $\widetilde{U}$ to the membership space $\mathbb{I}$ (When $\mathbb{I}$ contains only the two points 0 and 1, $\mathbb{A}$ is nonfuzzy and $\pi_{\mathbb{A}}(\mathfrak{u})$ is identical to the characteristic function of a nonfuzzy set).The nonnegative real numbers with finite supremums are included in the range of the membership function. Typically, zero-degree members of elements are not listed.

Definition (2.2) [7]: $\mathcal{S}(\mathbb{A})$, is the crisp set of all $\mathfrak{u} \in \widetilde{U}$,the support of a fuzzy set $\mathbb{A}$ such that $\pi_{\mathbb{A}}(\mathfrak{u}) > 0$.

Definition (2.3) [7] : The elements that belong the fuzzy set's (crisp) $\mathbb{A}$ The r-level set is defined as at least to the degree r. $\mathbb{A}_r = \left\{\mathfrak{u} \in \widetilde{U} : \pi_{\mathbb{A}}(\mathfrak{u}) \ge r\right\}$

$\grave{\mathbb{A}}_r = \left\{\mathfrak{u} \in \widetilde{U} : \pi_{\mathbb{A}}(\mathfrak{u}) > r\right\}$ is called a "strong r-level set" or a "strong r-cut".

Definition (2.4) [8]: A fuzzy set $\mathbb{A}$ is convex if

$$\mathbb{A}(\alpha \mathfrak{u}_1 + (1-\alpha)\mathfrak{u}_2) \ge min\{\pi_{\mathbb{A}}(\mathfrak{u}_1), \pi_{\mathbb{A}}(\mathfrak{u}_2)\}, \mathfrak{u}_1, \mathfrak{u}_2 \in \widetilde{U}, \alpha \in [0,1]$$

Alternatively, if all r -level sets are convex, then a fuzzy set is convex.

Definition (2.5) [9]: Fuzzy Number

An ordered pair of functions $\overline{p}(r)$,$\underline{p}(r)$, $0 \le r \le 1$ completely determines a fuzzy number p, which satisfy the following requirements:

1) $\underline{p}(r)$ is a bounded left continuous and non-decreasing function on [0,1].

2) $\overline{p}(r)$ is a bounded right continuous and non-increasing function on [0,1].

3) $\underline{p}(r) \le \overline{p}(r)$ $0 \le r \le 1$. The crisp number ($\mathfrak{a}$) is simply represented by:

$\underline{p}(r) = \overline{p}(r) = \mathfrak{a}$, $0 \le r \le 1$. The set of all the fuzzy numbers is denoted by $\mathbb{E}^1$.

Remark (2.1) [9]: For arbitrary $p = (\underline{p}, \overline{p})$, $q = (\underline{q}, \overline{q})$ and $\mathbb{C} \in$ R, the addition and multiplication by $\mathbb{C}$ can be defined as :

1) $(\underline{p+q})(r) = \underline{p}(r) + \underline{q}(r)$

2) $(\overline{p+q})(r) = \overline{p}(r) + \overline{q}(r)$

3) $(\underline{\mathbb{C}p})(r) = \mathbb{C}\underline{p}(r)$, $(\overline{\mathbb{C}p})(r) = \mathbb{C}\overline{p}(r)$, if $\mathbb{C} \geq 0$

4) $(\underline{\mathbb{C}p})(r) = \mathbb{C}\overline{p}(r)$ $(\overline{\mathbb{C}p})(r) = \mathbb{C}\underline{p}(r)$, if $\mathbb{C} < 0$. For all $r \in [0,1]$.

Definition (2.6) [9] : Fuzzy Function

A fuzzy function is the function ϑ: R $\longrightarrow \mathbb{E}^1$. We defined to each function $\mathbb{A}_1 \subseteq \mathbb{E}^1$ to $\mathbb{A}_2 \subseteq \mathbb{E}^1$ a fuzzy function.

Definition (2.7) [10]: The fuzzy function ϑ: R $\longrightarrow \mathbb{E}^1$ is considered to be continuous if:
$|\mathfrak{u} - \mathfrak{u}_1| < \delta \Rightarrow \mathbb{d}(\vartheta(\mathfrak{u}), \vartheta(\mathfrak{u}_1)) < \varepsilon$, for an arbitrary $\mathfrak{u}, \mathfrak{u}_1 \in$ R and $\varepsilon > 0$ there exists $\delta > 0$ where $\mathbb{d}$ is the distance between two fuzzy numbers.

3. Architecture Neuro – Fuzzy System

To show the structure of the Neuro – Fuzzy System (NFS) by converting real inputs ($x_1, x_2, \dots, x_i, \dots, x_n$) into fuzzy outputs $([\psi_1]_r, [\psi_2]_r, \dots, [\psi_k]_r, \dots, [\psi_s]_r)$ throughout the m hidden fuzzy neurons $([\mathrm{Hid}_1]_r, [\mathrm{Hid}_2]_r, \dots [\mathrm{Hid}_j]_r, \dots [\mathrm{Hid}_m]_r)$ such that $r \in [0,1]$. Let $[b_j]_r$ and $[\vartheta_k]_r$ are the fuzzy biases for the fuzzy neurons $[\mathrm{Hid}_j]_r$, $[\psi_k]_r$ resbictively , $[w_{ji}]_r$ is the fuzzy weight connecting x_i crisp neuron to $[\mathrm{Hid}_j]_r$ fuzzy neuron , and $[s_{kj}]_r$ is the fuzzy weight connecting fuzzy neuron $[\mathrm{Hid}_j]_r$ to fuzzy neuron $[\psi_k]_r$.

Input units: $x = x_i$, i=1,2,3,…,n (1)

Hidden units: $[\mathrm{Hid}_j]_r = T([\mathrm{Netw}_j]_r)$, j =1,2,3,…,m (2)

Where $[\mathrm{Netw}_j]_r = \sum_{i=1}^{n} x_i [w_{ji}]_r + [b_j]_r$ (3)

Output units : $[\psi_k]_r = T([Netw_k]_r)$, k=1,2,3,…,l (4)

Where $[Netw_k]_r = \sum_{j=1}^{m} [s_{kj}]_r [Hid_i]_r + [\vartheta_k]_r$ (5)

To use NFS to solve any fuzzy Singular Perturbation Problem (FSPP) for ODE. We use a multi-layer network with one unit entry (x), one hidden layer (m hidden neurons), and one linear output unit (i.e. the dimension of NFS). a multi-layer network having one unit entry x , one hidden layer with m hidden units (neurons) and one linear output unit, i.e. the dimension of NFS is $(1 \times m \times 1)$ as shown in (Fig. 1).

Input units: $x = x$ (6)

Hidden units : $[Hid_j]_r = [\overline{Hid_j}, \underline{Hid_j}] = [T([\overline{Netw_j}]), T([\underline{Netw_j}])]$ (7)

Where

$[\overline{Netw_j}] = x[\overline{w_j}] + [\overline{b_j}]$ (8)

$[\underline{Netw_j}] = x[\underline{w_j}] + [\underline{b_j}]$ (9)

Output units: $[\psi]_r = [\overline{\psi}, \underline{\psi}]$ (10)

Where $[\overline{\psi}] = \sum_{j \in a} [\overline{s_j}] [\overline{Hid_j}] + \sum_{j \in b} [\overline{s_j}] [\underline{Hid_j}]$ (11)

$\left[\underline{\psi}\right]=\sum_{j\in c}\left[\underline{s}_j\right]\left[\underline{Hid}_j\right]+\sum_{j\in d}\left[\underline{s}_j\right]\left[\overline{Hid}_j\right]$ (12)

For $\left[\underline{Hid}_j\right]\geq\left[\overline{Hid}_j\right]\geq 0$, where

a=$\{j;[\overline{s}_j]\geq 0\}$, b=$\{j;[\overline{s}_j]<0\}$ c=$\{j;[\underline{s}_j]\geq 0\}$, d=$\{j;[\underline{s}_j]<0\}$

$a\cup b=\{1,2,3,\ldots,m\}$,$c\cup d=\{1,2,3,\ldots,m\}$.

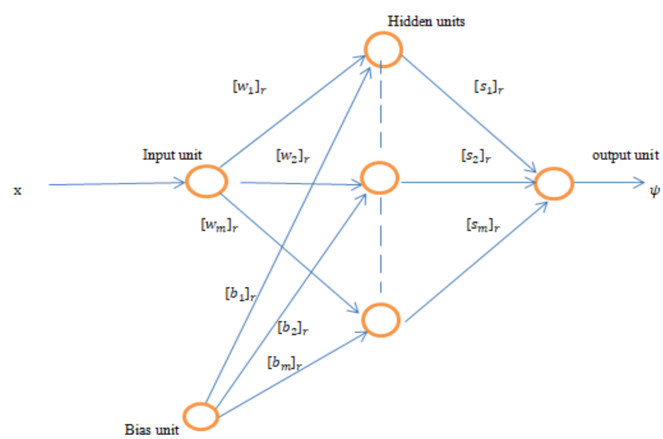

Fig. 1 - (1×m×1) dimension of NFS.

4. The Suggested Method

Numerous researchers have sought to create heuristic methods based on research the characteristics of conventional learning algorithms. These studies investigate concepts such as carefully choosing an activation function and using momentum to calculate the learning rate, various acceleration techniques had been developed. This method was utilized by Ezadi and Parandin to first-order ordinary differential equations are resolved in [10].In This method has been expanded in this chapter to solve FSPPs of the first and higher orders for ordinary differential equations.

This new approach relies on changing each x in the input vector (training set)
$\vec{x}=(x_1,x_2,\ldots,x_n)$,$x_j\in[a,b]$ a polynomial of first degree which will be as follows.

$\xi(x)=\frac{\lambda}{2}(\vec{x}+1)$, $\lambda\in(a,b)$ (13)

Then the input vector will be:

$(\xi(x_1),\xi(x_2),\ldots,\xi(x_n))$, $\xi(x_j)\in(a,b)$ and j=1,2,…,n (14)

The neural network cannot be trained in the first- and end-point range by selecting training points over the open interval(a,b)using MNFS. As a result, there is a reduction in the amount of computation that involves mistake. In reality, the training points for the neural network are transformed into similar points in the open interval(a,b) based on the distance $[a,b]$ chosen for training. The network is trained in these related domains by utilizing the new methodology. As seen above, we have:

For a given input vector$(x_1,x_2,\ldots,x_n)$, $x_j\in[a,b]$ and j =1,2,…,n (15)

The output of the MNFS is:

$[\psi]_r=\sum_{i=1}^{m}[s_i]_r\ T([Netw_i]_r)$ (16)

For i=1,…, m is the total number of hidden units
where

$[Netw_i]_r=\sum_{j=1}^{n}\left[w_{ij}\right]_r\ \xi(x_j)+[b_i]_r$ (17)

And $\xi(x_j) = \frac{\lambda}{2}(x_j + 1)$, $\lambda \in (0,1)$ (18)
where $x_j \in [a , b]$ and $\xi(x_j) \in (a , b)$, j=1,2,…,n
Note: For MNFS:

$$\frac{d[\psi]_r}{d[s_i]_r} = \sum_{i=1}^{m} T\left(\sum_{j=1}^{n} [w_{ij}]_r\ \xi(x_j) + [b_i]_r\right) = \sum_{i=1}^{h} T\left(\sum_{j=1}^{n} \frac{\lambda}{2}(x_j+1)[w_{ij}]_r + [b_i]_r\right) \quad (19)$$

$$\frac{d[\psi]_r}{d[b_i]_r} = \sum_{i=1}^{m} [s_i]_r T'\left(\sum_{j=1}^{n} [w_{ij}]_r\ \xi(x_j) + [b_i]_r\right) = \sum_{i=1}^{h} T'\left(\sum_{j=1}^{n} \frac{\lambda}{2}(x_j+1)[w_{ij}]_r + [b_i]_r\right) \quad (20)$$

$$\frac{d[\psi]_r}{d[w_{ij}]_r} = \sum_{i=1}^{m} [s]_r T'\left(\sum_{j=1}^{n} [w_{ij}]_r\ \xi(x_j) + [b_i]_r\right) = \sum_{i=1}^{h} \frac{\lambda}{2}(x_j+1) T'\left(\sum_{j=1}^{n} \frac{\lambda}{2}(x_j+1)[w_{ij}]_r\ [b_i]_r\right) \quad (21)$$

where T' is the activation function's first derivative.
After that, we showed how to select the right value for λ.

Theorem (2.1) : If $x \in [a, b]$ and a, b positive real number then we can find the suitable value of λ so as to ensure that $\xi(x) \in (a,b)$ such that: $\frac{2a}{a+1} < \lambda < \frac{2b}{b+1}$.
Proof: -Since $x \in [a , b]$, and since $\xi(x_j) = \frac{\lambda}{2}(x + 1)$,
Then we get: $\xi(x) = \frac{\lambda}{2}[a + 1, b + 1] = [\frac{\lambda}{2}(a + 1), \frac{\lambda}{2}(b + 1)]$
If we consider $\frac{\lambda}{2}(a + 1) = a$, then we get: $\lambda = \frac{2a}{a+1}$
And if we consider $\frac{\lambda}{2}(b + 1) = b$, then we get: $\lambda = \frac{2b}{b+1}$
Therefore, if we consider $\lambda > \frac{2a}{a+1}$ and $\lambda < \frac{2b}{b+1}$
We get: $\xi(x) = [\frac{\lambda}{2}(a + 1), \frac{\lambda}{2}(b + 1)] \in \left(\frac{a}{a+1}(a + 1), \frac{b}{b+1}(b + 1)\right) = (a, b)$
Therefore, we must have $\xi(x) \in (a,b)$ if $\frac{2a}{a+1} < \lambda < \frac{2b}{b+1}$.
When applicable, we narrow the interval from [a,b] to [0,b]
Therefore, we have $x \in [0, b]$ and we must have $\xi(x) \in (0,b)$ if suitable, value of δ is $0 < \lambda < \frac{2b}{b+1}$ Proving it is simple.

5. Illustration of MNFS for Solving FSPP With ICS.

5.1 Solution first-order of FSPP :

For illustration this steps, we will consider the first order of FSPPs for ODE with I.C

$$\varepsilon \frac{d\psi(x)}{dx} = F(x, \psi, \varepsilon) \ , \ x \in [a, b], 0 < \varepsilon \ll 1 \quad (22)$$

using the fuzzy initial condition $\psi(a) = A$, where $F(x , \psi(x), \varepsilon)$ is a fuzzy function of the crisp variable x and the fuzzy variable ψ, where ψ is a fuzzy function of x. while ψ' is the fuzzy derivative of ψ and A is a fuzzy number in E^1 under the r −cut sets , i.e. $[A]_r = [\overline{A}, \underline{A}]$, $r \in [0,1]$.
It is clear that the fuzzy function $f(x , \psi)$ is the mapping $f: R \times E^1 \longrightarrow E^1$
Now it is possible to replace (22) by the following equivalent system:
It is obvious that the fuzzy function $f(x , \psi)$is the mapping $f: R \times E^1 \longrightarrow E^1$

This equivalent system can now be used in place of (22):

$$\begin{cases} \varepsilon \frac{d\underline{\psi}(x)}{dx} = \underline{F}(x, \psi, \varepsilon) = G_1\left(x, \underline{\psi}, \overline{\psi}, \varepsilon\right), \underline{\psi}(a) = \underline{A} \\ \varepsilon \frac{d\overline{\psi}(x)}{dx} = \overline{F}(x, \psi, \varepsilon) = G_2\left(x, \underline{\psi}, \overline{\psi}, \varepsilon\right), \overline{\psi}(a) = \overline{A} \end{cases} \quad (23)$$

Where

$$\begin{cases} G_1\left(x, \underline{\psi}, \overline{\psi}, \varepsilon\right) = \text{Min}\left\{F(x,u) : u \in \left[\underline{\psi}, \overline{\psi}\right]\right\} \\ G_2\left(x, \underline{\psi}, \overline{\psi}, \varepsilon\right) = \text{Max}\left\{F(x,u) : u \in \left[, \underline{\psi}, \overline{\psi}\right]\right\} \end{cases} \quad (24)$$

System (23), which has the following parametric form:

$$\begin{cases} \varepsilon \frac{d\underline{\psi}(x,r)}{dx} = G_1\left[x, \underline{\psi}(x,r), \overline{\psi}(x,r), \varepsilon\right], \underline{\psi}(a,r) = \underline{A}(r) \\ \varepsilon \frac{d\overline{\psi}(x,r)}{dx} = G_2\left[x, \underline{\psi}(x,r), \overline{\psi}(x,r), \varepsilon\right], \overline{\psi}(a,r) = \overline{A}(r) \end{cases} \quad (25)$$

When $x \in [a, b]$ and $r \in [0, 1]$. A collection of points x_i,where $i = 1,2,3, \ldots g$, and so forth, are now created by discretizing the interval [a,b]. The system (25) can therefore be expressed as follows for any $x_i \in [a, b]$:

$$\begin{cases} \varepsilon \frac{d\underline{\psi}(x_i,r)}{dx_i} - G_1\left[x_i, \underline{\psi}(x_i,r), \overline{\psi}(x_i,r), \varepsilon\right] = 0 \\ \varepsilon \frac{d\,\overline{\psi}(x_i,r)}{dx_i} - G_2\left[x_i, \underline{\psi}(x_i,r), \overline{\psi}(x_i,r), \varepsilon\right] = 0 \end{cases} \quad (26)$$

With the initial conditions: $\underline{\psi}(a, r) = \underline{A}(r), \overline{\psi}(a, r) = \overline{A}(r), r \in [0, 1]$.

We written as The sum of two terms in the system's trial solutions (25), which are this subsection (and later in this chapter) to make use of the function approximation capabilities of feed-forward neural networks (see eq. 38 and 39). The first term has no movable parameters and meets the beginning conditions and boundary requirements. The second term's differential equations must be solved using a feed-forward neural network that has been trained. Given that any function can be approximated with arbitrary precision using a multilayer perceptron with a single hidden layer, for the second equation in system (25) where $\underline{p}$ and $\overline{p}$ are adjustable parameters.

As a form of network architecture, the multilayer perceptron is employed. If the first equation in system (25) has a trial solution of $\underline{\psi}_t(x, r, \underline{p}, \varepsilon)$ and the second equation has a trial solution of $\overline{\psi}_t(x, r, \overline{p}, \varepsilon)$" where $\underline{p}$ and $\overline{p}$ are movable parameters.

Indeed, $\underline{\psi}_t(x, r, \underline{p}, \varepsilon)$ and $\overline{\psi}_t(x, r, \overline{p}, \varepsilon)$ are approximation of $\underline{\psi}(x, r)$ and $\overline{\psi}(x, r)$ respectively, then a discretize version of the system (25) can be converted to the following optimization problem:

$$\min_{\vec{p}} \sum_{i=1}^{g} \begin{pmatrix} \left(\underline{\psi}'_t(x_i, r, \underline{p}, \varepsilon) - \frac{1}{\varepsilon} G_1\left[x_i, \underline{\psi}_t(x_i, r, \underline{p}, \varepsilon), \overline{\psi}_t(x_i, r, \overline{p}, \varepsilon)\right]\right)^2 \\ + \left(\overline{\psi}'_t(x_i, r, \overline{p}, \varepsilon) - \frac{1}{\varepsilon} G_2\left[x_i, \underline{\psi}_t(x_i, r, \underline{p}, \varepsilon), \overline{\psi}_t(x_i, r, \overline{p}, \varepsilon)\right]\right)^2 \end{pmatrix} \quad (27)$$

$\Big($Here $\vec{p} = (\underline{p}, \overline{p})$ contains all adjustable parameters$\Big)$ subject to the initial conditions:

$\underline{\psi}_t(a, r, \underline{p}, \varepsilon) = \underline{\psi}_0(r)$, $\overline{\psi}_t(a, r, \overline{p}, \varepsilon) = \overline{\psi}_0(r)$.

One feed-forward neural network is used in each trial solution of $\underline{\psi}_t(a, r, \underline{p}, \varepsilon)$ and $\overline{\psi}_t(a, r, \overline{p}, \varepsilon)$ with the corresponding networks indicated by $\underline{Out}(\xi(x), \xi(r), \underline{p}, \varepsilon)$ and $\overline{Out}(\xi(x), \xi(r), \overline{p}, \varepsilon)$ and adjustable parameters $\underline{p}$ and $\overline{p}$ respectively.

A fuzzy trial solution ψ_t can be written as:

$$[\psi_t(x, p, \varepsilon)]_r = [A]_r + (x - a)[\text{Out}(\xi(x), \xi(r), p, \varepsilon)]_r \quad (28)$$

Where $\text{Out}(\xi(x), \xi(r), p, \varepsilon)$ is the output of the feed forward NFS with one input unit for x and paramter p.

The quantity of mistake that needs to be minimized is expressed as:

$$E(p) = \sum_{i=1}^{g} \left(\left[\overline{E(p)}\right]_{ir} + \left[\underline{E(p)}\right]_{ir}\right) \quad (29)$$

Where $\{x_i\}_{i=1}^{g}$ are discrete points $\in [a, b]$, $\left[\overline{E(p)}\right]_{ir}$ *and* $\left[\underline{E(p)}\right]_{ir}$ can be thought as the squared errors for the lower and upper boundaries of the r-level sets, respectively.

$$\left[\overline{E(p)}\right]_{ir} = \left[\frac{d\overline{\psi}_t(x_i, \overline{p}, r, \varepsilon)}{dx} - \frac{1}{\varepsilon} F\left[x_i, \overline{\psi}_t(x_i, \overline{p}, \varepsilon), \varepsilon\right]\right]^2 \quad (30)$$

$$\left[\underline{E(p)}\right]_{ir} = \left[\frac{d\underline{\psi}_t(x_i, \underline{p}, r, \varepsilon)}{dx} - \frac{1}{\varepsilon} F\left[x_i, \underline{\psi}_t\left(x_i, \underline{p}, \varepsilon\right), \varepsilon\right]\right]^2 \quad (31)$$

The first derivative of $[\psi]_r$ can be expressed simply

$$\frac{\partial\overline{\psi}_t}{\partial x}=\sum_a \bar{s}_j \frac{\partial\overline{Hid_j}}{\partial\overline{Netw_j}}\frac{\partial\overline{Netw_j}}{\partial x}+\sum_b \bar{s}_j \frac{\partial\underline{Hid_j}}{\partial\underline{Netw_j}}\frac{\partial\underline{Netw_j}}{\partial x} \qquad (32)$$

$$\frac{\partial\underline{\psi}_t}{\partial x}=\sum_c \underline{s}_j \frac{\partial\underline{Hid_j}}{\partial\underline{Netw_j}}\frac{\partial\underline{Netw_j}}{\partial x}+\sum_d \underline{s}_j \frac{\partial\overline{Hid_j}}{\partial\overline{Netw_j}}\frac{\partial\overline{Netw_j}}{\partial x} \qquad (33)$$

Where $a=\{j : \bar{s}_j \geq 0\}$, $b=\{j: \bar{s}_j < 0\}$, $c=\{j : \underline{s}_j \geq 0\}$, $d=\{j : \underline{s}_j < 0\}$, $a \cup b=\{1,2,3,\ldots m\}$ and $c \cup d=\{1,2,3,\ldots m\}$. Such that

$$\frac{\partial\overline{Netw_j}}{\partial x}=\overline{w}_j \qquad (34)$$

$$\frac{\partial\overline{Hid_j}}{\partial\overline{Netw_j}}=1-\left(\overline{Hid_j}\right)^2 \qquad (35)$$

$$\frac{\partial\underline{Netw_j}}{\partial x}=\underline{w}_j \qquad (36)$$

$$\frac{\partial\underline{Hid_j}}{\partial\underline{Netw_j}}=1-\left(\underline{Hid_j}\right)^2 \qquad (37)$$

To drive the minimized error function for (22):
We find From (28):

$$\overline{\psi}_t(x,\overline{p},r,\varepsilon)=\overline{A}+(x-a)(\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon)) \qquad (38)$$

$$\underline{\psi}_t\left(x,\underline{p},r,\varepsilon\right)=\underline{A}+(x-a)(\underline{Out}(\xi(x),\xi(r),\underline{p},\varepsilon)) \qquad (39)$$

Then we derive the fuzzy trail solution in equations (38) and (39) to substitute it in the two equations (30),(31)

$$\frac{\partial\overline{\psi}_t(x,\overline{p},r,\varepsilon)}{\partial x}=\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon)+(x-a)\frac{\partial\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon)}{\partial x} \qquad (40)$$

$$\frac{\partial\underline{\psi}_t(x,\underline{p},r,\varepsilon)}{\partial x}=\underline{Out}(\xi(x),\xi(r),\underline{p},\varepsilon)+(x-a)\frac{\partial\underline{Out}(\xi(x),\xi(r),\underline{p},\varepsilon)}{\partial x} \qquad (41)$$

$$\left[\overline{E(p)}\right]_{ir}=\left[\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon)+(x_i-a)\frac{\partial\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon)}{\partial x}-\frac{1}{\varepsilon}F(x_i,\overline{A}+(x_i-a)\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon),\varepsilon)\right]^2 \qquad (42)$$

$$\left[\underline{E(p)}\right]_{ir}=\left[\underline{Out}(\xi(x),\xi(r),\underline{p},\varepsilon)+(x_i-a)\frac{\partial\underline{Out}(\xi(x),\xi(r),\underline{p},\varepsilon)}{\partial x}-\frac{1}{\varepsilon}F(x_i,\underline{A}+(x_i-a)(\underline{Out}(\xi(x),\xi(r),\underline{p},\varepsilon),\varepsilon\right]^2 \qquad (43)$$

$$\left[\overline{E(p)}\right]_{ir}=\left[\sum_a[\bar{s}_j][\overline{Hid_j}]+\sum_b[\bar{s}_j]\left[\underline{Hid_j}\right]+(x_i-a)[\bar{s}_j][\overline{w}_j]\left(1-([\overline{Hid_j}])^2\right)+[\bar{s}_j][\underline{w}_j]\left(1-\left(\left[\underline{Hid_j}\right]\right)^2\right)-\frac{1}{\varepsilon}F\left(x_i,\overline{A}+(x_i-a)\left(\sum_a[\bar{s}_j][\overline{Hid_j}]+\sum_b[\bar{s}_j]\left[\underline{Hid_j}\right]\right)\right),\varepsilon\right]^2 \qquad (44)$$

$$\left[\underline{E(p)}\right]_{ir}=\left[\sum_c[\underline{s}_j]\left[\underline{Hid_j}\right]+\sum_d[\underline{s}_j][\overline{Hid_j}]+(x_i-a)[\underline{s}_j][\underline{w}_j]\left(1-\left(\left[\underline{Hid_j}\right]\right)^2\right)+[\underline{s}_j][\overline{w}_j]\left(1-([\overline{Hid_j}])^2\right)-\frac{1}{\varepsilon}F\left(x_i,\underline{A}+(x_i-a)\left(\sum_c[\underline{s}_j]\left[\underline{Hid_j}\right]+\sum_d[\underline{s}_j][\overline{Hid_j}]\right)\right),\varepsilon\right]^2 \qquad (45)$$

where:
$a=\{j: [\bar{s}_j] \geq 0\}$, $b=\{j: [\bar{s}_j] < 0\}$, $c=\{j: [\underline{s}_j] \geq 0\}$,
$d=\{j: [\underline{s}_j] < 0\}$, $a \cup b=\{1,2,3,\ldots m\}$ and $c \cup d=\{1,2,3,\ldots m\}$

5.2 Solution second-order of FSPPs:

Now we will consider the second order of FSPPs

$$\varepsilon\frac{d^2\psi(x)}{dx^2}=F(x,\psi,\psi',\varepsilon) \quad x\in[a,b],\ 0<\varepsilon<<1 \qquad (46)$$

With initial conditions (IC): $\psi(a)=A$, $\psi(a)=A'$ where A and A' are a fuzzy numbers in E^1 under the r −cut sets, ε is perturbation parameter i.e.$0<\varepsilon\ll 1$.

$[A]_r=[\overline{A},\underline{A}],\ r\in[0,1]$

$[A']_r=\left[\overline{A'},\underline{A'}\right],\ r\in[0,1]$

A fuzzy trial solution ψ_t can written as:

$$[\psi_t(\mathrm{x},\mathrm{p},\varepsilon)]_\mathrm{r}=[A]_\mathrm{r}+[A']_\mathrm{r}(\mathrm{x}-a)+(\mathrm{x}-a)^2\,[\mathrm{Out}(\,\xi(\mathrm{x}),\xi(\mathrm{r}),\mathrm{p},\varepsilon)]_\mathrm{r} \quad (47)$$

From (47) we can find:

$$\overline{\psi}_t(\mathrm{x},\overline{p},\varepsilon)=\overline{A}+\overline{\acute{A}}(\mathrm{x}-a)+(\mathrm{x}-a)^2(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)) \quad (48)$$

$$\underline{\psi}_\mathrm{t}\left(\mathrm{x},\underline{p},\varepsilon\right)=\underline{A}+\underline{\acute{A}}\,(\mathrm{x}-a)+(\mathrm{x}-a)^2\,(\underline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)) \quad (49)$$

Where $\mathrm{Out}(\,\xi(\mathrm{x}),\xi(\mathrm{r}),\mathrm{p},\varepsilon)$ is the output of the feed forword NFS with one input unit for x and weight p.
The required minimum amount of error is expressed as:

$$\mathrm{E}(\mathrm{p})=\sum_{\mathrm{i}=1}^{\mathrm{g}}\left(\left[\overline{E(p)}\right]_{ir}+\left[\underline{E(p)}\right]_{ir}\right) \quad (50)$$

Where

$$\left[\overline{E(p)}\right]_{ir}=\left[\frac{\partial^2\overline{\psi}_t(\mathrm{x_i},\overline{p},\varepsilon)}{\partial \varkappa^2}-\frac{1}{\varepsilon}\left[F\left[\mathrm{x}_i,\overline{\psi}_t(\mathrm{x}_i,\overline{\mathrm{p}},\varepsilon),\frac{\partial\overline{\psi}_t(\mathrm{x_i},\overline{p},\varepsilon)}{\partial\mathrm{x}},\varepsilon\right)\right]\right]^2,\mathrm{x}_i\in[a,b] \quad (51)$$

$$\left[\underline{E(p)}\right]_{ir}=\left[\frac{\partial^2\underline{\psi}_\mathrm{t}\left(\mathrm{x_i},\underline{\mathrm{p}},\varepsilon\right)}{\partial\varkappa^2}-\frac{1}{\varepsilon}[F\left[\mathrm{x}_i,\underline{\psi}_\mathrm{t}\left(\mathrm{x}_i,\underline{p},\varepsilon\right),\frac{\partial\underline{\psi}_\mathrm{t}\left(\mathrm{x_i},\underline{p},\varepsilon\right)}{\partial\mathrm{x}},\varepsilon\right)\right]\right]^2,\ \mathrm{x}_i\in[a,b] \quad (52)$$

Since

$$\frac{\partial\overline{\psi}_t(\mathrm{x},\overline{p},\varepsilon)}{\partial\mathrm{x}}=\overline{\acute{A}}+(\mathrm{x}-a)^2\frac{\partial(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon))}{\partial\mathrm{x}}+2(\mathrm{x}-a)(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)) \quad (53)$$

$$\frac{\partial\underline{\psi}_\mathrm{t}\left(\mathrm{x},\underline{p},\varepsilon\right)}{\partial\mathrm{x}}=\underline{\acute{A}}+(\mathrm{x}-a)^2\,\frac{\partial(\underline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon))}{\partial\mathrm{x}}+2(\mathrm{x}-a)(\underline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)) \quad (54)$$

$$\frac{\partial^2\overline{\psi}_t(\mathrm{x},\overline{p},\varepsilon)}{\partial\varkappa^2}=(\mathrm{x}-a)^2\,\frac{\partial^2(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon))}{\partial\varkappa^2}+2(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon))\ +4(\mathrm{x}-a)\frac{\partial(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon))}{\partial\varkappa} \quad (55)$$

$$\frac{\partial^2\underline{\psi}_\mathrm{t}\left(\mathrm{x},\underline{p},\varepsilon\right)}{\partial\varkappa^2}=(\mathrm{x}-a)^2\,\frac{d^2(\underline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon))}{\partial\varkappa^2}+2\,(\underline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon))+4(\mathrm{x}-a)\frac{\partial(\underline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon))}{\partial\varkappa} \quad (56)$$

It is simple to assess the performance gradient in relation to the NFS coefficient. $\left[\overline{E(p)}\right]_{ir}=\Big[(\mathrm{x}-a)^2\,\frac{d^2(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)}{\partial\mathrm{x}^2}+2\left(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon\right)+4(\mathrm{x}-a)\frac{\partial\left(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon\right)}{\partial\mathrm{x}}-\frac{1}{\varepsilon}\left[F(\mathrm{x}_i,\overline{A}+\overline{\acute{A}}(\mathrm{x}-a)+(\mathrm{x}-a)^2(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon),\overline{\acute{A}}+(\mathrm{x}-a)^2\frac{\partial(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)}{\partial\mathrm{x}}+2(\mathrm{x}-a)(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)\varepsilon)\right]^2,\mathrm{x}_i\in[a,b]$ (57)

$$\left[\underline{\mathrm{E(p)}}\right]_{\mathrm{ir}}=\Big[(\mathrm{x}-\mathrm{a})^2\,\frac{\mathrm{d}^2\,(\underline{\mathrm{Out}}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)}{\partial\mathrm{x}^2}+2\,(\underline{\mathrm{Out}}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)+4(\mathrm{x}-\mathrm{a})\frac{\partial(\underline{\mathrm{Out}}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)}{\partial\mathrm{x}}-\frac{1}{\varepsilon}[\mathrm{F}(\mathrm{x_i},\underline{\mathrm{A}}+\underline{\acute{\mathrm{A}}}\,(\mathrm{x}-\mathrm{a})+(\mathrm{x}-\mathrm{a})^2\,(\underline{\mathrm{Out}}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon),\underline{\acute{\mathrm{A}}}+(\mathrm{x}-\mathrm{a})^2\,\frac{\partial(\underline{\mathrm{Out}}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)}{\partial\mathrm{x}}+2(\mathrm{x}-\mathrm{a})(\underline{\mathrm{Out}}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon),\varepsilon\Big]^2,\mathrm{x_i}\in[\mathrm{a},\mathrm{b}] \quad (58)$$

6. Numerical illustrations

We provide numerous we include various examples to demonstrate the behavior and efficacy of the suggested NFS. We use MATLAB version 7.12 to develop the programs, We propose a three-layer feed forward NN with a single input unit, a single hidden layer with seven ODE-specific hidden units, and a single linear output unit. The activation function of the hidden units is the hyperbolic tangent function, and its formula is: $T(\mathrm{x})\ =\ \frac{\mathrm{e^x-e^{-x}}}{\mathrm{e^x+e^{-x}}}$.

6.1. Example : Consider the first order of nonlinear FSPPs:

$$\varepsilon\,\psi'(\mathrm{x})=\frac{1}{2}A\psi^2,\qquad \mathrm{x}\in[0\,,0.1],\text{ and } r\in[0\,,1]$$

with the fuzzy initial conditions :

$[\psi(0)]_r=\left[0.5\sqrt{r}\,,0.2\sqrt{1-r}+0.5\right],\ A=[1+r,3-r]$

The fuzzy analytical solutions is:

$$[\psi_\mathrm{a}(\mathrm{x})]_r=\left[\frac{0.5\sqrt{r}}{1-3(1+r)(0.5\sqrt{r})\mathrm{x}},\frac{0.2\sqrt{1-r}+0.5}{1-3(3+r)(0.2\sqrt{1-r}+0.5)\mathrm{x}}\right]$$

Then fuzzy trial solutions

$[\psi_t(\mathrm{x},\mathrm{p})]_\mathrm{r}=\left[0.5\sqrt{r}\,,0.2\sqrt{1-r}+0.5\right]_\mathrm{r}+\mathrm{x}[\mathrm{Out}(\,\xi(\mathrm{x}),\xi(\mathrm{r}),\mathrm{p},\varepsilon)]_\mathrm{r}$

we have:

$\overline{\psi}_t(\mathrm{x},\overline{\mathrm{p}})=0.5\sqrt{r}+\mathrm{x}(\overline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon))$

$\underline{\psi}_t\left(\mathrm{x},\underline{\mathrm{p}}\right)=\ 0.2\sqrt{1-r}+0.5+\mathrm{x}(\underline{Out}(\xi(\mathrm{x})\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon))$

The MNFS was trained using a grid of 10 evenly spaced points in the range [0,0.1], the input vector x (training set) is as follows: $\vec{\mathrm{x}}=\{0,0.01,0.02\,,0.03,0.04,0.05,0.06\,,0.07,0.08\,,0.09\,,0.1\}$ We now take the following actions in order to identify E the error function that needs to be minimized for this issue:

$$\frac{d\overline{\psi}_t(\mathrm{x},\overline{\mathrm{p}})}{dx} = \overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon) + \mathrm{x}\frac{d(\overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon))}{dx}$$

$$\frac{d\underline{\psi}_t\left(\mathrm{x},\underline{\mathrm{p}}\right)}{dx} = \underline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon) + \mathrm{x}\frac{d(\underline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon))}{dx}$$

Then we get:

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[\frac{d\overline{\psi}_t(\mathrm{x}_i,\overline{\mathrm{p}})}{dx} - (\frac{1}{2\varepsilon}(1+r)\left[\overline{\psi_t}^2(\mathrm{x}_i)\right]\right]^2$$

$$\underline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[\frac{d\underline{\psi}_t\left(\mathrm{x}_i,\underline{\mathrm{p}}\right)}{dx} - (\frac{1}{2\varepsilon}(3-r)\left[\underline{\psi_t}^2(\mathrm{x}_i)\right]\right]^2$$

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[\overline{Out}(\xi(\mathrm{x}_i), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon) + \mathrm{x}\frac{d\overline{Out}(\xi(\mathrm{x}_i), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon)}{dx} - (\frac{1}{2\varepsilon}(1+r)[\,0.5\sqrt{r} + \mathrm{x}_i(\overline{Out}(\xi(\mathrm{x}_i), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon))]^2\right]^2$$

$$\underline{\mathbb{E}}_{\mathrm{ir}}(\mathrm{p},\varepsilon) = \left[\underline{\mathrm{Out}}(\xi(\mathrm{x}_i), \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon) + \mathrm{x}\frac{\mathrm{d}\underline{\mathrm{Out}}(\xi(\mathrm{x}_i), \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon)}{\mathrm{dx}} - (\frac{1}{2\varepsilon}(3-\mathrm{r})\left[0.2\sqrt{1-\mathrm{r}} + 0.5 + \mathrm{x}_i(\underline{\mathrm{Out}}(\xi(\mathrm{x}_i), (\mathrm{r}), \underline{\mathrm{p}}, \varepsilon))\right]^2\right]^2$$

Since

$$\overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon) = \sum_{\mathrm{j}=1}^{7} \overline{\mathrm{s}}_{\mathrm{j}}\, T\ \left(\xi(\mathrm{x})\overline{\mathrm{w}}_{\mathrm{j}} + \overline{\mathrm{b}}_{\mathrm{j}}\right)$$

$$\underline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon) = \sum_{\mathrm{j}=1}^{7} \underline{s_j}\, T\ \left(\xi(\mathrm{x})\underline{w_j} + \underline{b_j}\right)$$

$$\frac{d\overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon)}{dx} = \sum_{\mathrm{j}=1}^{7} \frac{\lambda}{2}[\overline{\mathrm{s}}_{\mathrm{j}}][\overline{\mathrm{w}}_{\mathrm{j}}]T'\left(\xi(\mathrm{x})[\overline{\mathrm{w}}_{\mathrm{j}}] + \overline{\mathrm{b}}_{\mathrm{j}}\right)$$

$$\frac{\mathrm{d}\underline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon)}{\mathrm{dx}} = \sum_{\mathrm{j}=1}^{7} \frac{\lambda}{2}\left[\underline{\mathrm{s}_{\mathrm{j}}}\right]\left[\underline{\mathrm{w}_{\mathrm{j}}}\right]\mathrm{T}'\left(\xi(\mathrm{x})\left[\underline{\mathrm{w}_{\mathrm{j}}}\right] + \underline{b_j}\right)$$

Since $T'(\mathrm{r}) = 1 - T^2(\mathrm{r})$ then we get:

$$\frac{d\overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon)}{dx} = \sum_{\mathrm{j}=1}^{7}\left(\frac{\lambda}{2}[\overline{\mathrm{s}}_{\mathrm{j}}][\overline{\mathrm{w}}_{\mathrm{j}}] - \frac{\lambda}{2}[\overline{\mathrm{s}}_{\mathrm{j}}][\overline{\mathrm{w}}_{\mathrm{j}}]\, T^2\left(\xi(\mathrm{x})\overline{\mathrm{w}}_{\mathrm{j}} + \overline{\mathrm{b}}_{\mathrm{j}}\right)\right)$$

$$\frac{\mathrm{d}\underline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon)}{\mathrm{dx}} = \sum_{\mathrm{j}=1}^{7} \frac{\lambda}{2}\left[\underline{\mathrm{s}_{\mathrm{j}}}\right]\left[\underline{\mathrm{w}_{\mathrm{j}}}\right] - \frac{\lambda}{2}\left[\underline{\mathrm{s}_{\mathrm{j}}}\right]\left[\underline{\mathrm{w}_{\mathrm{j}}}\right] T^2\left(\xi(\mathrm{x})\underline{\mathrm{w}_{\mathrm{j}}} + \underline{b_j}\right)$$

Therefore we have:

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[\sum_{\mathrm{j}=1}^{7} \overline{\mathrm{s}}_{\mathrm{j}}\, T\ \left(\xi(\mathrm{x}_i)\overline{\mathrm{w}}_{\mathrm{j}} + \overline{\mathrm{b}}_{\mathrm{j}}\right) + \mathrm{x}_i\left(\sum_{\mathrm{j}=1}^{7} \frac{\lambda}{2}[\overline{\mathrm{s}}_{\mathrm{j}}][\overline{\mathrm{w}}_{\mathrm{j}}]T'\left(\xi(\mathrm{x}_i)[\overline{\mathrm{w}}_{\mathrm{j}}] + \overline{\mathrm{b}}_{\mathrm{j}}\right) - (\frac{1}{2\varepsilon}(1+r)[\,0.5\sqrt{r} + \mathrm{x}_i\left(\sum_{\mathrm{j}=1}^{7} \overline{\mathrm{s}}_{\mathrm{j}}\, T\ \left(\xi(\mathrm{x}_i)\overline{\mathrm{w}}_{\mathrm{j}} + \overline{\mathrm{b}}_{\mathrm{j}}\right)\right]^2\right]^2$$

$$\underline{\mathbb{E}}_{\mathrm{i}\tau}(\mathrm{p},\varepsilon) = \left[\sum_{\mathrm{j}=1}^{7} \underline{\mathrm{s}_{\mathrm{j}}}T\ \left(\xi(\mathrm{x}_i)\underline{\mathrm{w}_{\mathrm{j}}} + \underline{b_j}\right) + \mathrm{x}\left(\sum_{\mathrm{j}=1}^{7} \frac{\lambda}{2}\left[\underline{\mathrm{s}_{\mathrm{j}}}\right]\left[\underline{\mathrm{w}_{\mathrm{j}}}\right] - \frac{\lambda}{2}\left[\underline{\mathrm{s}_{\mathrm{j}}}\right]\left[\underline{\mathrm{w}_{\mathrm{j}}}\right]T^2\left(\xi(\mathrm{x}_i)\underline{\mathrm{w}_{\mathrm{j}}} + \underline{b_j}\right) - (\frac{1}{2\varepsilon}(3-\mathrm{r})\left[0.2\sqrt{1-r} + 0.5 + \mathrm{x}_i\left(\sum_{\mathrm{j}=1}^{7} \underline{\mathrm{s}_{\mathrm{j}}}\, T\ \left(\xi(\mathrm{x}_i)\underline{\mathrm{w}_{\mathrm{j}}} + \underline{b_j}\right)\right]^2\right]^2$$

The error function for this issue has to be minimized, and it is:

$$\mathbb{E}(p) = \sum_{i=1}^{g}\left(\overline{\mathbb{E}}_{ir}(p) + \underline{\mathbb{E}}_{ir}(p)\right)$$

For this example since xϵ[0,0.1] ,we choose $\varepsilon = \frac{1}{3}$ and according to theorem (2.1) we must choose $0 < \lambda < \frac{0.2}{1.1}$

For $\lambda = 0.05$ the training set will be

$\vec{\mathrm{x}} = \{0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1\}$

$\xi(\mathrm{x})$: 0.025 0.0252 0.0255 0.0257 0.026 0.0262 0.0265 0.0267 0.027 0.0272 0.0275.

Evaluation of the performance gradient in relation to the coefficient is simple. A grid of evenly spaced points in [0,0.1] was used to train the feed-forward MNFS , (Fig. 2) displays the neural and analytical solution in the training set. Following that, an oral MNFS result, an exact solution, and train accuracy faults are provided in a table (1) and (2).

Table 1 - Analytic and MNFS solution of example 6.1, ε=1/3, r=0.1.

input	Analytic solution		Solution of MNFS $\psi_t(x)$	
X	$\underline{\psi}_a(x)$	$\overline{\psi}_a(x)$	$\underline{\psi}_t(x)$	$\overline{\psi}_t(x)$
0	0.158113883008419	0.689736659610103	0.158113883008419	0.689736659617632
0.01	0.158943210237318	0.737012715648167	0.158943210239985	0.737012715645543
0.02	0.159781283190158	0.791246476056235	0.159781283190084	0.791246476057765
0.03	0.160628240943156	0.854095911406907	0.160628240945542	0.854095911406654
0.04	0.161484225537063	0.927791221474676	0.161484225599854	0.927791221474998
0.05	0.162349382056581	1.015405020884110	0.162349382055432	1.015405020877640
0.06	0.163223858712343	1.121291574703180	0.163223858710091	1.121291574755420
0.07	0.164107806925553	1.251832850082450	0.164107806922198	1.251832850088750
0.08	0.165001381415397	1.416774366550270	0.165001381415994	1.416774366550430
0.09	0.165904740289324	1.631777380452670	0.165904740665438	1.631777380455540
0.1	0.166818045136303	1.923710662138970	0.166818086654331	1.923710655432980

Table 2 - Accuracy of solutions of example 6.1, ε=1/3, r=0.1.

The error $[\mathbf{E}(\mathbf{x})]_{\mathbf{r}} = \lvert[\boldsymbol{\psi}_{\mathbf{a}}(\mathbf{x})]_r - [\boldsymbol{\psi}_{\mathbf{t}}(\mathbf{x})]_{\mathbf{r}}\rvert$	
$[\underline{\mathbf{E}(\mathbf{x})}]_{\mathbf{r}}$	$[\overline{\mathbf{E}(\mathbf{x})}]_{\mathbf{r}}$
0	7.5292E-12
2.66731E-12	2.62412E-12
7.43849E-14	1.52967E-12
2.38595E-12	2.53464E-13
6.27911E-11	3.21632E-13
1.14922E-12	6.47238E-12
2.25192E-12	5.22424E-11
3.35482E-12	6.29963E-12
5.96828E-13	1.57652E-13
3.76115E-10	2.8717E-12
4.1518E-08	6.70599E-09
MSE=1.56717E-16	MSE=4.08847E-18

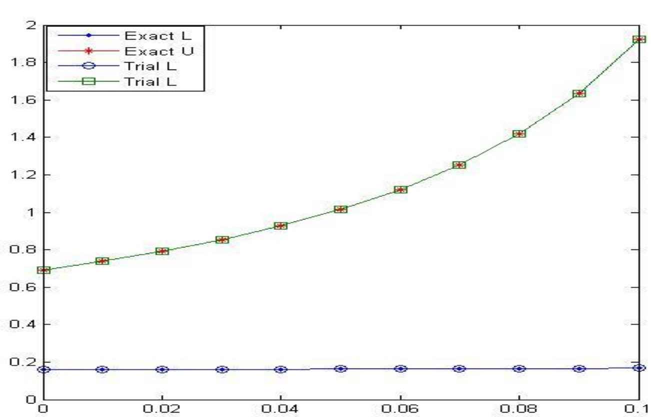

Fig. 1 - analytic and neurally solution of example 6.1, with ε=1/3.

6.2. Example: Consider the nonhomogeneous second order FDE:

$$\varepsilon\psi'' + \psi = \frac{1}{2}\cos x\ , \qquad x \in [0,1].$$

The fuzzy initial conditions
$[\psi(0)]_r = [2r\,, 4 - 2r],\quad [\psi'(0)]_r = [-2 + 2r\,, 2 - 2r],\ \ r \in [0,1].$
The fuzzy analytical solution is:
$[\psi_a(x)]_r = \left[\left(2r - \frac{2}{3}\right)\cos 2x + (r-1)\sin 2x + \frac{2}{3}\cos x\,, \left(\frac{10}{3} - 2r\right)\cos 2x + (1-r)\sin 2x + \frac{2}{3}\cos x\right]$
Then the fuzzy trial solution is
$[\psi_t(x,p)]_r = [2r\,, 4 - 2r]_r + [-2 + 2r\,, 2 - 2r\,]_r\, x + x^2[\mathrm{Out}(\xi(x), \xi(r), p, \varepsilon)]_r$.
Then we have:
$\overline{\psi_t}(x,\overline{p}) = 2r + [-2 + 2r]x + x^2(\overline{Out}(\xi(x)\,, \xi(r), \overline{p}, \varepsilon))$
$\underline{\psi_t}\left(x, \underline{p}\right) = 4 - 2r + [2 - 2r]x + x^2(\underline{Out}(\xi(x), \xi(r), \underline{p}, \varepsilon))$
The input vector $\vec{x}$ (training set) for the MNFS is a grid of ten evenly spaced points in the interval [0.1]:
$\vec{x} = \{0, 0.1, 0.2\ , 0.3, 0.4, 0.5, 0.6\ , 0.7, 0.8\ , 0.9\ , 1\}$.
Following are the steps we now take to determine the error function E that needs to be minimized for this problem:

$$\frac{d\overline{\psi_t}(x,\overline{p})}{dx} = -2 + 2r + x^2\frac{d(\overline{Out}(\xi(x)\,, \xi(r), \overline{p}, \varepsilon))}{dx} + 2x(\overline{Out}(\xi(x)\,, \xi(r), \overline{p}, \varepsilon)$$

$$\frac{d\underline{\psi_t}\left(x, \underline{p}\right)}{dx} = [2 - 2r] + x^2\frac{dx(\underline{Out}(\xi(x)\,, \xi(r), \underline{p}, \varepsilon))}{dx} + 2x(\underline{Out}(\xi(x)\,, \xi(r), \underline{p}, \varepsilon)$$

$$\frac{d^2\overline{\psi_t}(x,\overline{p})}{dx^2} = x^2\frac{d^2\overline{Out}(\xi(x), \xi(r), \overline{p}, \varepsilon)}{dx^2} + 2\left(\overline{Out}(\xi(x)\,, \xi(r), \overline{p}, \varepsilon\right) + 4x\frac{d\overline{Out}(\xi(x), \xi(r), \overline{p}, \varepsilon)}{dx})$$

$$\frac{d^2\,\underline{\psi_t}(x,\underline{p})}{dx^2} = x^2\frac{d^2\underline{Out}(\xi(x)\,, \xi(r), \underline{p}, \varepsilon)}{dx^2} + 2(\underline{Out}(\xi(x)\,, \xi(r), \underline{p}, \varepsilon) + 4x\frac{d\underline{Out}(\xi(x)\,, \xi(r), \underline{p}, \varepsilon)}{dx})$$

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[\frac{d^2\overline{\psi_t}(x_i, \overline{p})}{dx^2} - \frac{1}{\varepsilon}\left[\frac{1}{2}\cos x_i - (\overline{\psi_t}(x_i))\right]\right]^2$$

$$\underline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[\frac{d^2\,\underline{\psi_t}\left(x_i, \underline{p}\right)}{dx^2} - \frac{1}{\varepsilon}\left[\frac{1}{2}\cos x_i - (\underline{\psi_t}(x_i))\right]\right]^2$$

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon) = \Bigg[x_i^{\,2}\frac{d^2\overline{Out}(\xi(x_i), \xi(r), \overline{p}, \varepsilon)}{dx^2} + 2\left(\overline{Out}(\xi(x_i)\,, \xi(r), \overline{p}, \varepsilon\right) + 4x_i\frac{d\overline{Out}(\xi(x_i), \xi(r), \overline{p}, \varepsilon)}{dx}$$
$$- \frac{1}{\varepsilon}\left[\frac{1}{2}\cos x_i - (2r + [-2 + 2r]x_i + x_i^{\,2}(\overline{Out}(\xi(x_i)\,, \xi(r), \overline{p}, \varepsilon))\,)\right]\Bigg]^2$$

$$\underline{\mathbb{E}}_{ir}(p,\varepsilon) = \Bigg[x_i^{\,2}\frac{d^2\underline{Out}(\xi(x_i)\,, \xi(r), \underline{p}, \varepsilon)}{dx^2} + 2(\underline{Out}(\xi(x_i)\,, \xi(r), \underline{p}, \varepsilon) + 4x_i\frac{d\underline{Out}(\xi(x_i)\,, \xi(r), \underline{p}, \varepsilon)}{dx}$$
$$- \frac{1}{\varepsilon}\left[\frac{1}{2}\cos x_i - (4 - 2r) + [2 - 2r]x_i + x_i^{\,2}(\underline{Out}(\xi(x_i)\,, \xi(r), \underline{p}, \varepsilon))\right]\Bigg]^2$$

Since
$\overline{Out}(\xi(x)\,, \xi(r), \overline{p}, \varepsilon) = \sum_{j=1}^{7}\overline{s_j}\,T\ \left(\xi(x)\overline{w_j} + \overline{b_j}\right)$

$$\underline{Out}(\xi(x)\,, \xi(r), \underline{p}, \varepsilon) = \sum_{j=1}^{7}\underline{s_j}\,T\ \left(\xi(x)\underline{w_j} + \underline{b_j}\right)$$

$\frac{d\overline{Out}(\xi(x)\,, \xi(r), \overline{p}, \varepsilon)}{dx} = \sum_{j=1}^{7}\frac{\lambda}{2}[\overline{s_j}][\overline{w_j}\,]T'\left(\xi(x)[\overline{w_j}] + \overline{b_j}\right)$

$$\frac{d\underline{Out}(\xi(x)\,, \xi(r), \underline{p}, \varepsilon)}{dx} = \sum_{j=1}^{7}\frac{\lambda}{2}\left[\underline{s_j}\right]\left[\underline{w_j}\right]T'\left(\xi(x)\left[\underline{w_j}\right] + \underline{b_j}\right)$$

$\frac{d^2\overline{Out}(\xi(x), \xi(r), \overline{p}, \varepsilon)}{dx^2} = \sum_{j=1}^{7}\left(\left(\frac{\lambda}{2}\right)^2\overline{s_j}\,\overline{w_j}^{\,2}\ T''\ \left(\xi(x)\,\overline{w_j} + \overline{b_j}\right)\right)$

$$\frac{d^2\underline{Out}(\xi(x)\,,\xi(r),\underline{p},\varepsilon)}{dx^2}=\sum_{j=1}^{7}\left(\frac{\lambda}{2}\right)^2\underline{s_j}\,\underline{w_j}^2T''\left(\xi(x)\left[\underline{w_j}\right]+\underline{b_j}\right)$$

$T''=2\,T^3-2\,T$

$$\frac{d^2\,\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon)}{dx^2}=\sum_{j=1}^{7}\left(\left(\frac{\lambda}{2}\right)^2\overline{s_j}\,\overline{w_j}^2\,(2\,T^3-2\,T)\,\left(\xi(x)\,\overline{w_j}\,+\,\overline{b}_j\right)\right)$$

$$\frac{d^2\underline{Out}(\xi(x)\,,\xi(r),\underline{p},\varepsilon)}{dx^2}=\sum_{j=1}^{7}\left(\left(\frac{\lambda}{2}\right)^2\underline{s_j}\,\underline{w_j}^2(2\,T^3-2\,T)\,\left(\xi(x)\left[\underline{w_j}\right]+\underline{b_j}\right)\right)$$

Therefore we have

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon)=\Big[x_i^2\sum_{j=1}^{7}\left(\left(\frac{\lambda}{2}\right)^2\overline{s_j}\,\overline{w_j}^2\,(2\,T^3-2\,T)\,\left(\xi(x_i)\,\overline{w_j}\,+\,\overline{b}_j\right)\right)+2\sum_{j=1}^{7}\overline{s_j}\,T\,\left(\xi(x_i)\overline{w_j}+\overline{b_j}\right)\ +$$
$$4x_i\sum_{j=1}^{7}\frac{\lambda}{2}[\overline{s_j}][\overline{w_j}]T'\left(\xi(x_i)[\overline{w_j}]+\overline{b_j}\right)-\frac{1}{\varepsilon}\Big[\frac{1}{2}\cos x_i-(2r\ +[-2\ +2r]x_i+x_i^2(\sum_{j=1}^{7}\overline{s_j}\,T\,\left(\xi(x_i)\overline{w_j}+\overline{b_j}\right)\ )\Big]\Big]^2$$

$$\underline{\mathbb{E}}_{ir}(p,\varepsilon)=\Big[x_i^2\sum_{j=1}^{7}\left(\left(\frac{\lambda}{2}\right)^2\underline{s_j}\,\underline{w_j}^2(2\,T^3-2\,T)\,\left(\xi(x_i)\left[\underline{w_j}\right]+\underline{b_j}\right)\right)+2(\sum_{j=1}^{7}\underline{s_j}\,T\,\left(\xi(x_i)\underline{w_j}+\underline{b_j}\right)\ +$$
$$4x_i\sum_{j=1}^{7}\frac{\lambda}{2}\left[\underline{s_j}\right]\left[\underline{w_j}\right]T'\left(\xi(x_i)\left[\underline{w_j}\right]+\underline{b_j}\right)-\frac{1}{\varepsilon}\Big[\frac{1}{2}\cos x_i-(\,4-2r)+\ [2-2r]x_i+x_i^2(\sum_{j=1}^{7}\underline{s_j}\,T\,\left(\xi(x_i)\underline{w_j}+\underline{b_j}\right)))\Big]\Big]^2$$

For this issue, the error function that needs to be minimized is:

$$\mathbb{E}(p)=\sum_{i=1}^{g}\left(\overline{\mathbb{E}}_{ir}(p)+\underline{\mathbb{E}}_{ir}(p)\right)$$

For this example since $x\epsilon[0,1]$,we choose $\varepsilon=0.25$ and according to theorem (2.1) we must choose $0<\lambda<1$.
For $\lambda=0.5$ the training set will be
$\vec{x}=\{0, 0.1, 0.2\ ,0.3, 0.4, 0.5, 0.6\ ,0.7, 0.8\ ,0.9\ ,1\}$
$\xi(x)$: 0.25 0.27 0.3 0.32 0.35 0.37 0.4 0.42 0.45 0.47 0.5
Evaluation of the performance gradient in relation to the coefficient is simple. A grid of evenly spaced points in [0,1] was used to train the feed-forward MNFS , (Fig. 3)shows the training set's neural and analytical solution. Following that, an oral MNFS result, an exact solution, and train accuracy faults are provided in a table(3) and (4).

Table 3 - Analytic and MNFS solution of example 2, ε=0.25, r=0.4.

Input	**Analytic solution**		**Solution of MNFS $\psi_t(x)$**	
X	$\underline{\psi_a}(x)$	$\overline{\psi_a}(x)$	$\underline{\psi_t}(x)$	$\overline{\psi_t}(x)$
0	0.800000000000000	3.200000000000000	0.800000000000000	3.200000000000000
0.1	0.674810055420479	3.265373039193530	0.674810055420946	3.265373039188640
0.2	0.542534845709356	3.220383242086660	0.542534845709333	3.220383242085890
0.3	0.408150257368007	3.066526701225280	0.408150257368812	3.066526701229240
0.4	0.276521236041832	2.809444647554460	0.276521236048342	2.809444647591130
0.5	0.152212757824596	2.458703473677610	0.152212757824092	2.458703473699260
0.6	039314658956340	2.027420172861030	0.039314658956332	2.027420172892560
0.7	-0.058712760750052	1.531748058196680	-0.058712760750059	1.531748058196880
0.8	-0.139166291900298	0.990243178226415	-0.139166291900692	0.990243178226934
0.9	-0.200195545638886	0.423136584151539	-0.200195545638176	0.423136584151567
1	-0.240863163722935	-0.148458659245259	-0.240863163722371	-0.148458659245233

Table 4 - Accuracy of example 6.2 solutions , ε=0.25, r=0.4.

The error $[\mathbf{E(x)}]_r = \lvert[\boldsymbol{\psi}_a(\mathbf{x})]_r - [\boldsymbol{\psi}_t(\mathbf{x})]_r\rvert$	
$[\underline{\mathbf{E(x)}}]_r$	$[\overline{\mathbf{E(x)}}]_r$
0	0
4.66516E-13	4.89209E-12
2.25375E-14	7.70051E-13
8.05411E-13	3.9635E-12
6.51018E-12	3.66742E-11
5.0393E-13	2.16529E-11
7.47319E-15	3.15321E-11
7.19563E-15	2.0095E-13
3.93935E-13	5.18918E-13
7.09766E-13	2.77001E-14
5.63938E-13	2.55906E-14
MSE=4.04367E-24	MSE=2.5897E-22

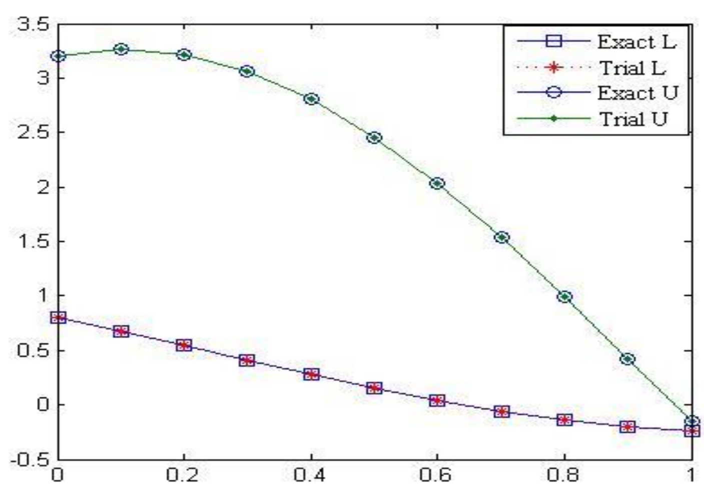

Fig. 3 - Analytic and neurally solution of example 6.2, with ε=0.25.

References

[1] Hristev R. M ,"The ANN Book", Edition 1, GNU Public License, ver. 2, 1998.
[2] C. Vikas , "Analysis of Back Propagation Algorithm", Vol. 2 , Issue 8, August 2013.
[3] J.S.R.Jang,Adaptivenetwork basedfuzzy inference systems,IEEE Transactions on Systems Man and Cybernetics 665–685,1993
[4] Shuai Song,Baoyong Zhang,Xiaona Song,Yijun Zhang,Zhengqiang Zhang,Wenjie Li. Fractional-order adaptive neuro-fuzzy sliding mode H ∞ control for fuzzy singularly perturbed systems, Journal of the Franklin Institute, Elsevier, July 2019, https://doi.org/10.1016/j.jfranklin.2019.03.020.
[5] K.M.M. Al-Abrahemee. Modification of high performance training algorithm for solving singular perturbation partial differential equations with cubic convergence, Journal of Interdisciplinary Mathematics, 24(7), 2035-2047, (2021) https://doi.org/10.1080/09720 502.2021.2001136
[6] Russul Kareem & Khalid M.M. Al-Abrahemee, Modification artificial neural networks for solving singular perturbation problems'Journal of Interdisciplinary Mathematics,(2022), https://doi.org/10.1080/09720502.2022.2072063.
[7] Hans-Jiirgen, "Fuzzy set theory and its applieations" , fourth ed,1934.
[8] Hamid. A. J., Rabha. W. I., Shayma A. M., Amera. I. M., and Samir. B. H., "Numerical solution of Lane-Emden equation using neural network", international conference on fundamental and applied sciences, vol. 1482, pp: 414- 418, 2012.
[9] Rawaa F.K. & Khalid M.M. Al-Abrahemee," Neuro-fuzzy system for solving fuzzy singular perturbation problems ",Journal of Interdisciplinary Mathematics,(2022), https://doi.org/10.1080/09720502.2022.2079229.
[10] S. Effati, M. Pakdaman, "Artificial Neural Network Approach for Solving Fuzzy Differential Equations", Information Sciences, 180, 1434-1457, 2010.