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)

Modification Fuzzy Artificial Neural Networks For Solving Fuzzy Singular Perturbation Problems With Boundary 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: 11 /01/2022
Revised form: 19 /02/2023
Accepted: 21 /02/2023
Available online: 24 /02/2023

Keywords:

Singular perturbation problem.
Neuro – fuzzy system.
Minimized Error Function.
Hyperbolic Tangent Activation Function.

ABSTRACT

Throughout this work through the using of a neuro-fuzzy system, we have developed a new technique. This updated approach is known neuro – fuzzy system method (MNFS). to develop a numerical method for resolving (FSPPs) for ordinary differential equations with BC. The activation function for hyperbolic tangents used to determine the hidden units' sigmoid function and the parameters of a fuzzy neural network and its formula is: $T(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$.Standard training algorithms and analytical techniques were contrasted with the suggested strategy. Our research revealed the provided approach stands out for its excellent accuracy of the results, low error rate, and much faster speed than that of conventional methods. A number of examples are used to demonstrate the suggested strategy.

MSC..

https://doi.org/10.29304/jqcm.2023.15.1.1149

1. Introduction

Nowadays the use of fuzzy singularly perturbed differential equations to describe problems in science and engineering has made this a topic that many scholars are interested in studying. In order to tackle the majority of practical issues since the solution of an FSPEs satisfies the fuzzy boundary requirements It is necessary to solve a fuzzy boundary issue . several issues fuzzy boundary value problems were difficult to answer precisely, and in certain cases, it was even impossible to come up with an analytical solution. So, it becomes more crucial to take into account their approximations, has been published a relatively recent study on the numerical solution of FSPEs using MNFS. The purpose of this study is to introduce a novel method for solving numerically fuzzy singularly perturbed differential equations with boundary conditions based on fuzzy neural networks. Artificial neural networks (ANNs), sometimes known as neural networks (NNs), more commonly, artificial neural networks (ANNs), refer to a group of

Corresponding author Tabark Aqeel Al-Janabi

Email addresses: tabark.aqeel98@gmail.com

Communicated by ‘sub etitor’

nonlinear computational techniques that, The original artificial neural networks (ANNs) were actually just integrated circuits designed to mimic and comprehend how nerve inputs and signals are transmitted in the central nervous system of humans. Therefore, throughout the past few years, there have been two main pathways taken by ANN research. On the other hand, computational NN has been steadily becoming popular as tools that can accomplish functions or solve issues that were thought to be challenging, it may be difficult or even impossible, For conventional mathematical and statistical methods. One may be said to concentrates on creating as accurate in silico human brain models as possible to understand more about all of its systems of behavior. It is more neuro physiologically oriented. The other views NNs solely as a computational tool for handling challenging issues, which are typically very nonlinear. neuro-fuzzy systems have piqued the interest of academics in a variety of scientific and technical fields because of its powerful learning and reasoning skills [1]. Artificial neural networks' learning capabilities are combined with fuzzy inference systems' explicit knowledge representation in neuro-fuzzy systems. according to the categorization of research articles from 2000 to 2017, this report suggests an overview of various neuro-fuzzy systems. This survey's major goal is to give readers a comprehensive understanding of the state of the art in neuro-fuzzy systems so simply techniques based on their research interests. Different neuro-fuzzy models are compared with their applications [2]. Artificial neural networks (ANNs) have recently been used for the estimate of the ordinary differential equation (ODE) and partial differential equation (PDE). We Several publications in the literature about differential equations are briefly reviewed in [3] two-point boundary value issues can be solved quickly using FFNN. Hussein in [4] developed FFNN to address singular boundary value issues. Tawfiq and Al-Abrahemee in [5] created an ANN to address issues with singular perturbation, and other researchers. In fact, NNs are being used in every circumstance where there are issues with prediction, categorization, or control. A few important reasons are responsible for this enormous accomplishment. First and foremost, ANNs are highly developed nonlinear computational tools that can simulate incredibly complex functions. An appropriately selected ANN architecture can reflect any specific functional connection between a set of inputs and matching outputs. The dimensionality problem7, which plagues attempts to represent nonlinear functions with a lot of variables, is likewise controlled inside this framework by NNs. Additionally, NNs learn by doing: Through the use of judiciously crafted training algorithms. When compared to, say, some more conventional nonlinear statistical methods, the level of user knowledge necessary to implement NNs successfully is substantially lower[6,7]. However, in order to pick and prepare data, choose the proper NN, and interpret the findings, the user does require some heuristic knowledge.

2. Basic definitions

We start first with the original definition of fuzzy sets which was first initiated by Zadeh in 1965.

Definition (2.1) [8]: Assume that X is a classical collection of items known as the universal set, where x stands for the generic elements. In many cases, the membership in a classical subset $\mathbb{A}$ of X is seen as a characteristic function $U_{\mathbb{A}}$ from into {0, 1}, such that:

$$U_{\mathbb{A}}(x) = \begin{cases} 1 & if\ x \in \mathbb{A} \\ 0 & if\ x \in \mathbb{A} \end{cases}$$

A valuation set is {0, 1}. A is known as a fuzzy set (which is denoted by $\mathbb{A}$) if it is allowed for the valuation set to be the real interval [0, 1], and $U_{\tilde{\mathbb{A}}}(x)$ is the grade of membership of x in $\tilde{\mathbb{A}}$.

Additionally, it is remarkable that the closer the value of $U_{\tilde{\mathbb{A}}}(x)$ to 1, the more x belong to $\tilde{\mathbb{A}}$ and vise versa. It is obvious that $\tilde{\mathbb{A}}$ is a subset of X that lacks a sharp boundary. Moreover, the fuzzy set $\tilde{\mathbb{A}}$ is described as: $\tilde{\mathbb{A}} = \{(x, U_{\tilde{\mathbb{A}}}(x)) : x \in \mathrm{X}, 0 \le U_{\tilde{\mathbb{A}}}(x) \le 1\}$

Remark (2.1) [9]:

Let X be the universal set and $\tilde{\mathbb{A}}$ be a fuzzy subset of X:

1. The elements of X, such that $U_{\tilde{\mathbb{A}}}(x) = \frac{1}{2}$ are called the crossover points of $\tilde{\mathbb{A}}$.
2. The height of $\tilde{\mathbb{A}}$ is the greatest membership value, i.e., $\mathrm{hgt}(\tilde{\mathbb{A}}) = \mathrm{Sup}_{x \in X} U_{\tilde{\mathbb{A}}}(x)$
3. $\tilde{\mathbb{A}}$ is said to be normal if and only if there exists $x \in X$ such that $U_{\tilde{\mathbb{A}}}(x) = 1$, otherwise $\tilde{\mathbb{A}}$ is subnormal.
4. The empty fuzzy set $\emptyset$ with membership function $U_{\tilde{\emptyset}}(x) = 0$, $\forall x \epsilon X$.
5. A non-empty fuzzy set $\tilde{\mathbb{A}}$ can always normalized by dividing $U_{\tilde{\mathbb{A}}}(x)$, $\forall x \in X$ by $\mathrm{Sup}_{x \in X} U_{\tilde{\mathbb{A}}}(x)$. We will typically assume that fuzzy sets are normalized.
6. The support of a fuzzy set $\tilde{\mathbb{A}}$ (denoted by supp ($\tilde{\mathbb{A}}$)) is the crisp set of all $x \in X$, such that $U_{\tilde{\mathbb{A}}}(x) > 0$.
7. A fuzzy set $\tilde{\mathbb{A}}$ is convex on $\mathfrak{U}$ if and only if: $U_{\tilde{\mathbb{A}}}(\delta x_1 + (1-\delta)x_2) \ge Min\{U_{\tilde{\mathbb{A}}}(x_1), U_{\tilde{\mathbb{A}}}(x_2)\}$ for all $x_1 , x_2 \in \mathfrak{U}$, and all $\delta \in [0, 1]$.

The r-Level sets

This section's focus is on the fundamental and crucial characteristics of so-called r-level sets, which are very significant in fuzzy sets. As a collection that connects fuzzy sets and regular sets, r-level sets are used, This can be used to demonstrate that certain results that are satisfied in regular sets are likewise satisfied in fuzzy sets, i.e., The regular sets and fuzzy sets can also be connected to one another in a different way.

Definition (2.2) [10]:

The crisp set of all x in X is known as the r-level or (r-cut) set of a fuzzy set $\widetilde{\mathbb{A}}$, abbreviated as $\mathbb{A}_r$ such that $U_{\widetilde{\mathbb{A}}}(x) \geq r$ i.e.,

$$\mathbb{A}_r = \{x \in X | U_{\widetilde{\mathbb{A}}}(x) \geq r, r \in (0,1]\}$$

Remark (2.2) [11]:

The strong r-level sets can be defined as follows:

$$\mathbb{A}_r{}^{+} = \{x \in X | U_{\widetilde{\mathbb{A}}}(x) > r, r \in (0,1]\}$$

The fulfillment of the ensuing characteristics for $r, t \in (0,1]$:can be simply verified:

1. $(p \cup q)_r = p_r \cup q_r$
2. $(p \cap q)_r = p_r \cap q_r$
3. $\tilde{p} \subseteq \tilde{q}$ $gives$ $p_r \subseteq q_r$
4. If $r \leq t$, then $p_r \supseteq p_t$
5. $\tilde{p} = \tilde{q}$ if and only if $p_r = q_r, \forall r \in (0,1]$
6. $p_r \cap p_t = p_t$ and $p_r \cap p_t = p_r$, if $r \leq t$.
7. If $\widetilde{\mathbb{A}}$ is a fuzzy set, $\{\mathbb{A}_r\}, \forall r \in (0,1]$ is a family of subsets of the universal set X, then:

$$\widetilde{\mathbb{A}} = \bigcup_{r \in [0,1]} r\mathbb{A}_r$$

Accordingly, as illustrated graphically in (Fig. 1)Each fuzzy set's r-levels collectively make up a family of nested crisp sets.

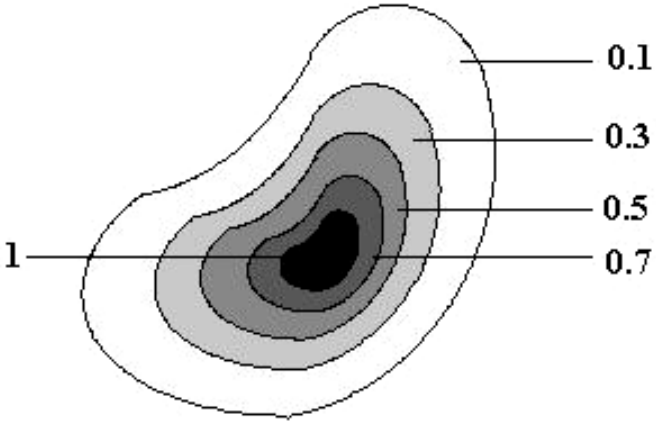

Fig. 1 - Nesting r-level sets.

3. Architecture Neuro – Fuzzy System

In order to demonstrate the construction of the Neuro-Fuzzy System (NFS), real inputs (x_1 , x_2, … , x_i , … , x_n)are transformed into fuzzy outputs ($[\psi_1]_r$, $[\psi_2]_r$, … , $[\psi_k]_r$, … , $[\psi_s]_r$)by the m hidden fuzzy neurons $\left([\mathrm{Hid}_1]_r , [\mathrm{Hid}_2]_r , \ldots [\mathrm{Hid}_j]_r , \ldots [\mathrm{Hid}_m]_r\right)$ such that $r \in [0 , 1]$. For the fuzzy neurons $[\mathrm{Hid}_j]_r$, $[\psi_k]_r$ there are fuzzy biases $[b_j]_r$ and $[\vartheta_k]_r$ respectively, let $[w_{ji}]_r$ the fuzzy weight linking x_i to fuzzy neuron$[\mathrm{Hid}_j]_r$, and $[s_{kj}]_r$ the fuzzy weight connecting between $[\mathrm{Hid}_j]_r$ fuzzy neuron to$[\psi_k]_r$ fuzzy neuron.

Input units: $x=x_i$, i=1,2,3,…,n (1)

Hidden units: $[\mathrm{Hid_j}]_r=T\left([\mathrm{Netw_j}]_r\right)$, 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 solve any FSPP (fuzzy singular perturbation problem) for an ODE. By use NFS As demonstrated in (Fig. 2), we employ a multi-layer network with one unit entry x, one hidden layer with m hidden units (neurons), and one linear output unit

Input units: $x=x$ (6)

Hidden units : $[Hid_j]_r=\left[\overline{Hid_j},\underline{Hid_j}\right]=\left[T([\overline{Netw_j}]),T([\underline{Netw_j}])\right]$ (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: $[\psi]_r=\left[\overline{\psi},\underline{\psi}\right]$ (10)

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

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

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

$a=\{j;[\overline{s_j}]\geq 0\}$, $b=\{j;[\overline{s_j}]<0\}$

$c=\{j;[\underline{s_j}]\geq 0\}$, $d=\{j;[\underline{s_j}]<0\}$, $a\cup b=\{1,2,3,\dots,m\}$, $c\cup d=\{1,2,3,\dots,m\}$

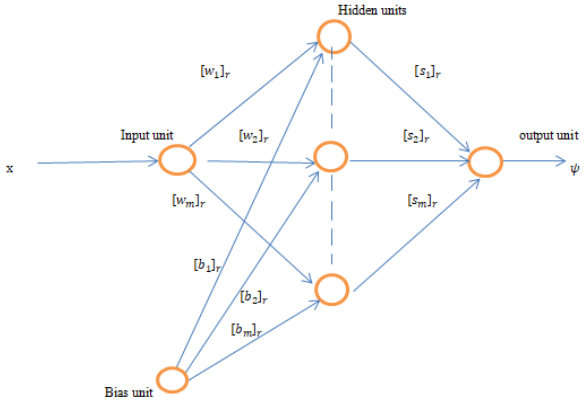

Fig. 2 - (1×m×1) dimension of NFS.

4. The Suggested Method

On the basis of study into the properties of traditional learning algorithms, many academics have attempted to develop heuristic techniques. These studies investigate concepts like carefully selecting an activation function and computing the learning rate using momentum. The heuristic technique made use of various acceleration mechanisms that had been developed. This method was utilized by Ezadi and Parandin in [12] to solve first order ODEs. The method for solving FSPPs of the first and higher orders for ordinary differential equations is developed in this chapter.

The primary component of this novel method is the replacement of each x in the input vector with a first-degree polynomial. (training set) $\vec{x}=(x_1,x_2,\dots,x_n)$,$x_j\in[a,b]$

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

Then the input vector will be:

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

When employing MNFS, selecting training points throughout the open interval (a, b) avoids training the neural network in the first and end point range. As a result, the amount of calculation that involves inaccuracy in computation is decreased. In reality, utilizing the novel method, the training points for the neural network that are dependent on the distance [a, b] are transformed into comparable points in the open interval (a , b), after which the network is trained in these similar locations.

Looking down, we have: $(x_1 , x_2 , \dots , x_n) , x_j \in [a, b]$ 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

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

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

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

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

T' the first derivative of the activation function.

5. Illustration of MNFS for Solving FSPP With BC.

Consider the second order of FSPP for ODEs

$$\varepsilon \frac{d^2\psi}{dx^2} = F\left(x, \psi(x), \frac{d\psi}{dx}, \varepsilon\right), \qquad x \in [0,1] \qquad (22)$$

where ε is perturbation paremter $(0 < \varepsilon << 1)$

With the fuzzy (BC): $\psi(a) = [A]_r$, $\psi(b) = [B]_r$

where $[A]_r$ and $[B]_r$ are fuzzy number in E^1 with $r-$ level sets

$[A]_r = [\overline{A}, \underline{A}]$ *and* $[B]_r = [\overline{B}, \underline{B}]$

Can be written a fuzzy trial solution:

$[\psi_t(x, p, \varepsilon)]_r = \frac{b[A]_r - a[B]_r}{b-a} + \frac{[B]_r - [A]_r}{b-a} x + (x-a)(x-b)[\text{Out}(\xi(x), \xi(r), p, \varepsilon)]_r$ (23)

where Out $(\xi(x), \xi(r), p, \varepsilon)$ is output of the feed forword NFS with one input for x and parametr p.

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

$\mathbb{E}(p) = \sum_{i=1}^{g} \left(\overline{\mathbb{E}}_{ir}(p) + \underline{\mathbb{E}}_{ir}(p)\right)$ (24)

Where

$$\overline{\mathbb{E}}_{ir}(p, \varepsilon) = \left[\frac{d^2\overline{\psi}_t(x_i, \overline{p}, \varepsilon)}{dx^2} - \frac{1}{\varepsilon}\left[F\left[x_i, \overline{\psi}_t(x_i, \overline{p}, \varepsilon), \frac{d\overline{\psi}_t(x_i, \overline{p}, \varepsilon)}{dx}, \varepsilon\right]\right]\right]^2 \qquad (25)$$

$$\underline{\mathbb{E}}_{ir}(p, \varepsilon) = \left[\frac{d^2\underline{\psi}_t(x_i, \underline{p})}{dx^2} - \frac{1}{\varepsilon}\left[F\left[x_i, \underline{\psi}_t(x_i, \underline{p}), \frac{d\underline{\psi}_t(x_i, \underline{p})}{dx} \varepsilon\right]\right]\right]^2 \qquad (26)$$

$\overline{\mathbb{E}}_{ir}$ and $\underline{\mathbb{E}}_{ir}$ are squared errors for the lower and upper limits of the r - level sets, respectively.Where $\{x_i\}_{i=1}^{g}$ the points discrete$[a, b]$.

Then (25) and (26) can be rewritten as:

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon)=\left[(\mathrm{x}_i-\mathrm{a})(\mathrm{x}_i-\mathrm{b})\frac{d^2(\overline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)}{d\mathrm{x}^2}+2\big(2\mathrm{x}_i-(a+b)\big)\frac{d(\overline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)}{d\mathrm{x}}+2(\overline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)-\frac{1}{\varepsilon}F\left(\mathrm{x}_i,\frac{b\overline{A}-a\overline{B}}{b-a}+\frac{\overline{B}-\overline{A}}{b-a}\mathrm{x}_i+(\mathrm{x}_i-a)(\mathrm{x}_i-b)(\overline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)_r,\frac{\overline{B}-\overline{A}}{b-a}+(\mathrm{x}_i-a)(\mathrm{x}_i-b)\frac{d(\overline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)}{d\mathrm{x}}+\big(2\mathrm{x}_i-(a+b)\big)(\overline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)\big)\right]^2 \quad (27)$$

$$\underline{\mathbb{E}}_{ir}(p,\varepsilon)=\left[(\mathrm{x}_i-\mathrm{a})(\mathrm{x}_i-\mathrm{b})\frac{d^2(\underline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)}{d\mathrm{x}^2}+2\big(2\mathrm{x}_i-(a+b)\big)\frac{d(\underline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)}{d\mathrm{x}}+2(\underline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)-\frac{1}{\varepsilon}F(\mathrm{x}_i,\frac{b\underline{A}-a\underline{B}}{b-a}+\frac{\underline{B}-\underline{A}}{b-a}\mathrm{x}_i+(\mathrm{x}_i-a)(\mathrm{x}_i-b)(\underline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon),\frac{\underline{B}-\underline{A}}{b-a}+(\mathrm{x}_i-a)(\mathrm{x}_i-b)\frac{d(\underline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)}{d\mathrm{x}}\big(2\mathrm{x}_i(a+b)\big)\big)(\underline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)\right]^2 \quad (28)$$

6. Numerical results

In this section, we give examples of MATLAB 7.12 programs that exhibit In order to demonstrate the technique the characteristics and behavior of the proposed design will be discussed. In order to solve FSPPs for ODEs, a multi-layer fuzzy feed forward FFNN with one input unit, one hidden layer with seven hidden units (neurons), and one linear output unit was utilized. Tansig is for each buried unit activating sigmoidly. We demonstrate some numerical experiments using the NFS, and we found that the suggested technique significantly increases accuracy and speeds up convergence while reducing the quantity of iterations necessary to achieve the goal.

6.1. Example : Consider the second-order linear FDE:

$$\varepsilon\psi''-0.5(\psi'+\psi)=0 \quad \mathrm{x}\in[0,1],\ \varepsilon \text{ is pertuerbation paramtor}$$

The fuzzy boundary conditions

$[\psi(0)]_r=\left[-\sqrt{-0.02\log_e r},\sqrt{-0.02\log_e r}\right]$

$[\psi(1)]_r=\left[14.60\sqrt{-2\log_e r},14.83\sqrt{-2\log_e r}\right]$

Where the fuzzy exact solution for this problem is:

$$[\psi_a(\mathrm{x})]_r=\left[2e^{2t}t-e^{-2t}\left(\frac{1}{10}\right)\sqrt{-2\log_e r}-e^{-2t}t\left(\frac{3}{10}\right)\sqrt{-2\log_e r}\cos(4t),2e^{2t}+e^{-2t}\left(\frac{1}{10}\right)\sqrt{-2\log_e r}+e^{-2t}t\left(\frac{3}{10}\right)\sqrt{-2\log_e r}\cos(4t)\right]$$

Then a fuzzy trial solutions is :

$[\psi_t(\mathrm{x},p)]_r=\left[-\sqrt{-0.02\log_e r},\sqrt{-0.02\log_e r}\right]_r+\left[14.60\sqrt{-2\log_e r},14.83\sqrt{-2\log_e r}\right]_r-\left[-\sqrt{-0.02\log_e r},\sqrt{-0.02\log_e r}\right]_r+\mathrm{x}(\mathrm{x}-1)[\mathrm{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\mathrm{p},\varepsilon)]_r$

Then we have:

$\overline{\psi_t}(\mathrm{x},\overline{\mathrm{p}})=-\sqrt{-0.02\log_e r}+14.60\sqrt{-2\log_e r}-\left[-\sqrt{-0.02\log_e r}\right]+\mathrm{x}(\mathrm{x}-1)\overline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)$

$\underline{\psi_t}\left(\mathrm{x},\underline{\mathrm{p}}\right)=\sqrt{-0.02\log_e r}+14.83\sqrt{-2\log_e r}-\left[\sqrt{-0.02\log_e r}\right]+\mathrm{x}(\mathrm{x}-1)\underline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)$

The MNFS a grid of ten evenly spaced spots was used to train in the interval [0,1], (i.e.) the input vector $\vec{\mathrm{x}}$(training set) is:

$\vec{\mathrm{x}}=\{0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1\}$.

The next step is to discover It is necessary to decrease the error function E for this issue:

$$\frac{d\overline{\psi_t}(\mathrm{x},\overline{\mathrm{p}})}{d\mathrm{x}}=(\mathrm{x}^2-\mathrm{x})\frac{d\overline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)}{d\mathrm{x}}+\overline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\overline{\mathrm{p}},\varepsilon)\ (2\mathrm{x}-1)$$

$$\frac{d\underline{\psi_t}\left(\mathrm{x},\underline{\mathrm{p}}\right)}{d\mathrm{x}}=(\mathrm{x}^2-\mathrm{x})\frac{d\underline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)}{d\mathrm{x}}+\underline{Out}(\xi(\mathrm{x}),\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)\ (2\mathrm{x}-1)$$

$$\frac{d^2\overline{\psi_t}(x,\overline{p})}{dx^2} = (x^2 - x)\frac{d^2\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon)}{dx^2} + 2\left[(2x-1)\frac{d\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon)}{dx}\right] + 2\left[\overline{Out}(\xi(x),\xi(r),\overline{p},\varepsilon)\right]$$

$$\frac{d^2\,\underline{\psi_t}(x,\underline{p})}{dx^2} = (x^2 - x)\frac{d^2\underline{Out}(\xi(x),\xi(r),\underline{p},\varepsilon)}{dx^2} + 2\left[(2x-1)\frac{d\underline{Out}(\xi(x),\xi(r),\underline{p},\varepsilon)}{dx}\right] + 2\left[\underline{Out}(\xi(x),\xi(r),\underline{p},\varepsilon)\right]$$

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[\frac{d^2\overline{\psi_t}(x_i,\overline{p})}{dx^2} - \frac{1}{\varepsilon}\left[0.5(\overline{\psi_t}'(x_i) + \overline{\psi_t}(x_i))\right]\right]^2$$

$$\underline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[\frac{d^2\,\underline{\psi_t}(x_i,\underline{p})}{dx^2} - \frac{1}{\varepsilon}\left[0.5(\underline{\psi_t}'(x_i) + \underline{\psi_t}(x_i))\right]\right]^2$$

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[(x_i{}^2 - x_i)\frac{d^2\overline{Out}(\xi(x_i),\xi(r),\overline{p},\varepsilon)}{dx^2} + 2\left[(2x_i-1)\frac{d\overline{Out}(\xi(x_i),\xi(r),\overline{p},\varepsilon)}{dx}\right] + 2\left[\overline{Out}(\xi(x_i),\xi(r),\overline{p},\varepsilon)\right] - \frac{1}{\varepsilon}\left[0.5\left((x_i{}^2 - x_i)\frac{d\overline{Out}(\xi(x_i),\xi(r),\overline{p},\varepsilon)}{dx} + \overline{Out}(\xi(x_i),\xi(r),\overline{p},\varepsilon)(2x_i-1) + (-\sqrt{-0.02\log_e r} + 14.60\sqrt{-2\log_e r} - \left[-\sqrt{-0.02\log_e r}\right] + x_i(x_i-1)\overline{Out}(\xi(x_i),\xi(r),\overline{p},\varepsilon)\right)\right]\right]^2$$

$$\underline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[(x_i{}^2 - x_i)\frac{d^2\underline{Out}(\xi(x_i),\xi(r),\underline{p},\varepsilon)}{dx^2} + 2\left[(2x_i-1)\frac{d\underline{Out}(\xi(x_i),\xi(r),\underline{p},\varepsilon)}{dx}\right] + 2\left[\underline{Out}(\xi(x_i),\xi(r),\underline{p},\varepsilon)\right] - \frac{1}{\varepsilon}\left[0.5\left((x_i{}^2 - x_i)\frac{d\underline{Out}(\xi(x_i),\xi(r),\underline{p},\varepsilon)}{dx} + \underline{Out}\left(\xi(x_i),\xi(r),\underline{p},\varepsilon\right)(2x_i-1) + \sqrt{-0.02\log_e r} + 14.83\sqrt{-2\log_e r} - \sqrt{-0.02\log_e r} + x_i(x_i-1)\underline{Out}(\xi(x_i),\xi(r),\underline{p},\varepsilon)\right)\right]^2$$

Since

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

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

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

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

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

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

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

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

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

Therefore we have:

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[(x_i{}^2 - x_i)\sum_{j=1}^{7}\left(\left(\frac{\lambda}{2}\right)^2\overline{s_j}\,\overline{w_j}^2\,(2\,T^3 - 2\,T)\left(\xi(x_i)\,\overline{w_j} + \overline{b_j}\right)\right) + 2\left[(2x_i-1)\sum_{j=1}^{7}\frac{\lambda}{2}\left[\overline{s_j}\right]\left[\overline{w_j}\right]T'\left(\xi(x_i)\left[\overline{w_j}\right] + \overline{b_j}\right)\right] + 2\left[\sum_{j=1}^{7}\overline{s_j}\,T\,\left(\xi(x_i)\overline{w_j} + \overline{b_j}\right)\right] - \frac{1}{\varepsilon}\left[0.5\left((x_i{}^2 - x_i)\sum_{j=1}^{7}\frac{\lambda}{2}\left[\overline{s_j}\right]\left[\overline{w_j}\right]T'\left(\xi(x_i)\left[\overline{w_j}\right] + \overline{b_j}\right) + \sum_{j=1}^{7}\overline{s_j}\,T\,\left(\xi(x_i)\overline{w_j} + \overline{b_j}\right)(2x_i-1) + (-\sqrt{-0.02\log_e r} + 14.60\sqrt{-2\log_e r} - \left[-\sqrt{-0.02\log_e r}\right] + x_i(x_i-1)\sum_{j=1}^{7}\overline{s_j}\,T\,\left(\xi(x_i)\overline{w_j} + \overline{b_j}\right)\right)\right]\right]^2$$

$$\underline{\mathbb{E}}_{ir}(p,\varepsilon) = \left[(x_i{}^2 - x_i)\sum_{j=1}^{7}\left(\left(\frac{\lambda}{2}\right)^2\underline{s_j}\,\underline{w_j}^2(2\,T^3 - 2\,T)\left(\xi(x_i)\left[\underline{w_j}\right] + \underline{b_j}\right)\right)2\left[(2x_i-1)\sum_{j=1}^{7}\frac{\lambda}{2}\left[\underline{s_j}\right]\left[\underline{w_j}\right]T'\left(\xi(x_i)\left[\underline{w_j}\right] + \underline{b_j}\right)\right] + 2\left[\sum_{j=1}^{7}\underline{s_j}\,T\,\left(\xi(x_i)\underline{w_j} + \underline{b_j}\right)\right] - \frac{1}{\varepsilon}\left[0.5\left((x_i{}^2 - x_i)\sum_{j=1}^{7}\frac{\lambda}{2}\left[\underline{s_j}\right]\left[\underline{w_j}\right]T'\left(\xi(x_i)\left[\underline{w_j}\right] + \underline{b_j}\right) + \sum_{j=1}^{7}\underline{s_j}\,T\,\left(\xi(x_i)\underline{w_j} + \underline{b_j}\right)(2x_i-1) + \sqrt{-0.02\log_e r} + 14.83\sqrt{-2\log_e r} - \left[-\sqrt{-0.02\log_e r}\right] + x_i(x_i-1)\sum_{j=1}^{7}\underline{s_j}\,T\,\left(\xi(x_i)\underline{w_j} + \underline{b_j}\right)\right)\right]^2$$

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 ϵ[0,1] , we choose $\varepsilon = 0.12$ and according to theorem (2.1) we must choose $0 < \lambda < \frac{4}{3}$
For $\lambda = 0.2$ 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
ξ(x):0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.2
The performance gradient in relation to the coefficient is simple to assess.
An equal equidistant grid of points in [0, 1], (Fig. 3) shows the analytic and neurally solution in the training set, which was utilized to train the feed forward NN. Then oral result of MNFS, Table (1) and (2) both provide the precise answer and train's accuracy errors.

Table 1 - Accuracy of solutions of example 1, ε=0.125,r=0.8.

input	Analytic solution		Solution of MNFS $\boldsymbol{\psi_t(x)}$	
x	$\underline{\boldsymbol{\psi}_a}(x)$	$\overline{\boldsymbol{\psi}_a}(x)$	$\underline{\boldsymbol{\psi}_t}(x)$	$\overline{\boldsymbol{\psi}_t}(x)$
0	-0.066804723083658	0.066804723083658	-0.066804723083658	0.066804723083644
0.1	0.149930578003846	0.292137789226413	0.149930578743290	0.292137789276640
0.2	0.416912231179531	0.560209975348605	0.416912288749650	0.560209956287580
0.3	0.740255186734336	0.879575382356868	0.740255177510600	0.879575382358831
0.4	1.127421704666120	1.259497811559920	1.127421705439860	1.259497811557640
0.5	1.587281060211060	1.710161481189200	1.587281066509440	1.710161877548740
0.6	2.130203211889260	2.242881909047960	2.130203266544460	2.242881909047720
0.7	2.768184880601560	2.870322700315780	2.768760163876540	2.870322700317710
0.8	3.515007506268910	3.606723464906990	3.515007576043750	3.606723464912870
0.9	4.386427438203900	4.468143761961120	4.386427438861350	4.468143761976380
1	5.400399512437810	5.472727801398370	5.400399512437630	5.472727801396120

Table 2 - Analytic and MNFS solution of example 6.1, ε=0.125, r=0.8.

The error $[\mathbf{E(x)}]_r = \lvert[\boldsymbol{\psi}_a(x)]_r - [\boldsymbol{\psi}_t(x)]_r\rvert$	
$[\underline{\mathbf{E(x)}}]_r$	$[\overline{\mathbf{E(x)}}]_r$
0	1.34476E-14
7.39444E-10	5.02265E-11
5.75701E-08	1.9061E-08
9.22374E-09	1.96299E-12
7.73743E-10	2.27529E-12
6.29838E-09	3.9636E-07
5.46552E-08	2.37144E-13
0.000575283	1.93445E-12
6.97748E-08	5.88329E-12
6.57455E-10	1.52554E-11
1.78524E-13	2.25153E-12
MSE=3.00864E-08	MSE=1.43149E-14

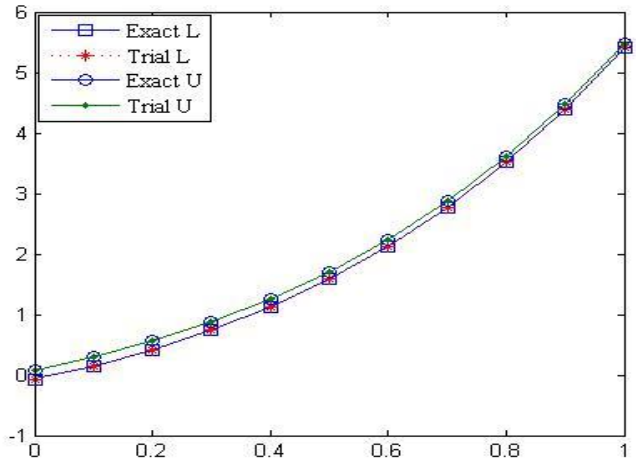

Fig. 3 - An analytic and neurally solution of example 6.1, with ε=0.125.

6.2. Example: Consider the second-order FDE :

$$\varepsilon\psi''(\mathrm{x}) = 2\psi^3 \ , \mathrm{x}\epsilon[0,1]$$

$$[\psi'(0)]_r = [1-r, r-1], [\psi'(1)]_r = \left[\frac{1}{4}(1-r), \frac{1}{4}(r-1)\right]$$

Where the fuzzy exact solution is :

$[\psi_a(\mathrm{x})]_r = \left[(r-1)\frac{1}{1+\mathrm{x}}, (1-r)\frac{1}{1+\mathrm{x}}\right]$.

Then a fuzzy trial solution is:

$[\psi_t(\mathrm{x},p)]_r = [1-r, r-1]_r + \left[\frac{1}{4}(1-r), \frac{1}{4}(r-1)\right]_r - [1-r, r-1]_r + \mathrm{x}(\mathrm{x}-1)[\mathrm{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \mathrm{p}, \varepsilon)]_r$

we have:

$$\overline{\psi_t}(\mathrm{x}, \overline{\mathrm{p}}) = 1 - r + \left[\frac{1}{4}(1-r)\right] - [1-r] + \mathrm{x}(\mathrm{x}-1)\overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon)$$

$$\underline{\psi_t}\left(\mathrm{x}, \underline{\mathrm{p}}\right) = r - 1 + \left[\frac{1}{4}(r-1)\right] - [r-1] + \mathrm{x}(\mathrm{x}-1)\underline{Out}(\xi(\mathrm{x}) , \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon)$$

The MNFS a grid of ten evenly spaced spots was used to train in the interval $[0,1]$, (i.e.) the input vector $\vec{\mathrm{x}}$(training set) is:

$\vec{\mathrm{x}} = \{0, 0.1, 0.2 , 0.3, 0.4, 0.5, 0.6 , 0.7, 0.8 , 0.9 , 1\}$.

The next step is to discover It is necessary to decrease the error function E for this issue:

$$\frac{d\overline{\psi_t}(\mathrm{x}, \overline{\mathrm{p}})}{d\mathrm{x}} = (\mathrm{x}^2 - \mathrm{x})\frac{d\overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon)}{d\mathrm{x}} + \overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon)\ (2\mathrm{x}-1)$$

$$\frac{d\underline{\psi_t}\left(\mathrm{x}, \underline{\mathrm{p}}\right)}{d\mathrm{x}} = (\mathrm{x}^2 - \mathrm{x})\frac{d\underline{Out}(\xi(\mathrm{x}) , \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon)}{d\mathrm{x}} + \underline{Out}(\xi(\mathrm{x}) , \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon)\ (2\mathrm{x}-1)$$

$$\frac{d^2\overline{\psi_t}(\mathrm{x}, \overline{\mathrm{p}})}{d\mathrm{x}^2} = (\mathrm{x}^2 - \mathrm{x})\frac{d^2\overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon)}{d\mathrm{x}^2} + 2\left[(2\mathrm{x}-1)\frac{d\overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon)}{d\mathrm{x}}\right] + 2\left[\overline{Out}(\xi(\mathrm{x}), \xi(\mathrm{r}), \overline{\mathrm{p}}, \varepsilon)\right]$$

$$\frac{d^2\underline{\psi_t}\left(\mathrm{x}, \underline{\mathrm{p}}\right)}{d\mathrm{x}^2} = (\mathrm{x}^2 - \mathrm{x})\frac{d^2\underline{Out}(\xi(\mathrm{x}) , \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon)}{d\mathrm{x}^2} + 2\left[(2\mathrm{x}-1)\frac{d\underline{Out}(\xi(\mathrm{x}) , \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon)}{d\mathrm{x}}\right] + 2\left[\underline{Out}(\xi(\mathrm{x}) , \xi(\mathrm{r}), \underline{\mathrm{p}}, \varepsilon)\right]$$

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

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

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

$$\left.\left.\overline{\mathrm{p}},\varepsilon)\right]-\frac{1}{\varepsilon}[2\left(1-r+\left[\frac{1}{4}(1-r)\right]-[1-r]+\mathrm{x}_i(\mathrm{x}_i-1)\overline{Out}(\xi(\mathrm{x}_i),\,\xi(\mathrm{r}),\,\overline{\mathrm{p}},\varepsilon)\right)^3\right]^2$$

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

$$\left.\left.\underline{\mathrm{p}},\varepsilon)\right]-\frac{1}{\varepsilon}[2\left(r-1+\left[\frac{1}{4}(r-1)\right]-[r-1]+\mathrm{x}_i(\mathrm{x}_i-1)\underline{Out}(\xi(\mathrm{x}_i)\,,\xi(\mathrm{r}),\underline{\mathrm{p}},\varepsilon)\right)^3\right]^2$$

Since

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

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

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

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

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

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

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

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

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

Therefore we have:

$$\overline{\mathbb{E}}_{ir}(p,\varepsilon)=\left[({\mathrm{x}_i}^2-\mathrm{x}_i)\sum_{\mathrm{j}=1}^{7}\left(\left(\frac{\lambda}{2}\right)^2\overline{\mathrm{s}_\mathrm{j}}\;\overline{w_\mathrm{j}}^2\;(2\,T^3-2\,T)\;\left(\xi(\mathrm{x}_i)\,\overline{w_\mathrm{j}}\;+\;\overline{\mathrm{b}_j}\right)\right)+2\left[(2\mathrm{x}_i-\right.\right.$$

$$1)\sum_{\mathrm{j}=1}^{7}\frac{\lambda}{2}\left[\overline{\mathrm{s}_\mathrm{j}}\right]\left[\overline{\mathrm{w}_\mathrm{j}}\right]T'\left(\xi(\mathrm{x}_i)\left[\overline{\mathrm{w}_\mathrm{j}}\right]+\overline{\mathrm{b}_\mathrm{j}}\right)\Big]+2\left[\sum_{\mathrm{j}=1}^{7}\overline{\mathrm{s}_\mathrm{j}}\;T\;\left(\xi(\mathrm{x}_i)\overline{\mathrm{w}_\mathrm{j}}+\overline{\mathrm{b}_\mathrm{j}}\right)\;\right]-\frac{1}{\varepsilon}[2\left(1-r+\left[\frac{1}{4}(1-r)\right]-\right.$$

$$\left.\left.[1-r]+\mathrm{x}_i(\mathrm{x}_i-1)\sum_{\mathrm{j}=1}^{7}\overline{\mathrm{s}_\mathrm{j}}\;T\;\left(\xi(\mathrm{x}_i)\overline{\mathrm{w}_\mathrm{j}}+\overline{\mathrm{b}_\mathrm{j}}\right)\;\right)^3\right]^2$$

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

$$1)\sum_{\mathrm{j}=1}^{7}\frac{\lambda}{2}\left[\underline{\mathrm{s}_\mathrm{j}}\right]\left[\underline{\mathrm{w}_\mathrm{j}}\right]\mathrm{T}'\left(\xi(\mathrm{x}_i)\left[\underline{\mathrm{w}_\mathrm{j}}\right]+\underline{b_j}\right)\Big]+2\left[\sum_{\mathrm{j}=1}^{7}\underline{s_j}\;T\;\left(\xi(\mathrm{x}_i)\underline{w_j}+\underline{b_j}\right)\right)\Big]-\frac{1}{\varepsilon}[2\left(r-1+\left[\frac{1}{4}(r-1)\right]-\right.$$

$$\left.\left.[r-1]+\mathrm{x}_i(\mathrm{x}_i-1)\sum_{\mathrm{j}=1}^{7}\underline{s_j}\;T\;\left(\xi(\mathrm{x}_i)\underline{w_j}+\underline{b_j}\right)\right)^3\right]^2$$

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 xϵ[0,1], we choose $\varepsilon=0.5$ and according to theorem (2.1) we must choose $0<\lambda<\frac{4}{3}$

For $\lambda=0.6$ 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
ξ(x): 0.3 0.33 0.36 0.39 0.42 0.45 0.48 0.51 0.54 0.57 0.6
The performance gradient in relation to the coefficient is simple to assess.
An equal equidistant grid of points in $[0, 1]$, (Fig. 4 & 5) shows the analytic and neurally solution in the training set, which was utilized to train the feed forward NN. Then oral result of MNFS, Table (3)(4)(5) and (6) both provide the precise answer and train's accuracy errors.

Table 3 - Analytic and MNFS solution of example 6.2, ε=0.5, r=0.7.

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	-0.300000000000000	0.300000000000000	-0.300000000000000	0.300000000000000
0.1	-0.272727272727273	0.272727272727273	-0.272727276784370	0.272727270653980
0.2	-0.250000000000000	0.250000000000000	-0.250000566532000	0.250004321998000
0.3	-0.230769230769231	0.230769230769231	-0.230769230553390	0.230769230769231
0.4	-0.214285714285714	0.214285714285714	-0.214285714299432	0.214285714867205
0.5	-0.200000000000000	0.200000000000000	-0.200004399875300	0.200004321998710
0.6	-0.187500000000000	0.187500000000000	-0.187500005439987	0.187500005422190
0.7	-0.176470588235294	0.176470588235294	-0.176470588994398	0.176470588999531
0.8	-0.166666666666667	0.166666666666667	-0.166666666677400	0.166666600418643
0.9	-0.157894736842105	0.157894736842105	-0.157894736800743	0.157894734219869
1	-0.150000000000000	0.150000000000000	-0.150000000000000	0.150000000000000

Table 4 - Accuracy of solutions of example 6.2, ε=0.5,r=0.7.

The error $[E(x)]_r = \lvert[\psi_a(x)]_r - [\psi_t(x)]_r\rvert$	
$[\underline{E(x)}]_r$	$[\overline{E(x)}]_r$
0	0
4.0571E-09	2.07329E-09
5.66532E-07	4.322E-06
2.15841E-10	0
1.37177E-11	5.81491E-10
4.39988E-06	4.322E-06
5.43999E-09	5.42219E-09
7.59104E-10	7.64237E-10
1.07333E-11	6.6248E-08
4.13623E-11	2.62224E-09
0	0
MSE=1.78908E-12	MSE=3.39671E-12

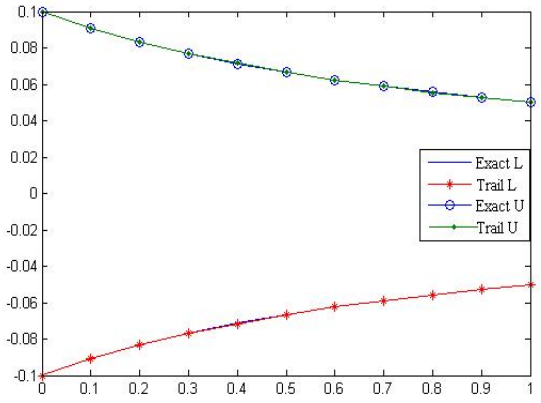

Fig. 4 - An analytic and neurally solution of example 6.2, with ε=0.5.

Table 5 - Analytic and MNFS solution of example 6.2, ε=0.5, x=0.5.

input	Analytic solution		Solution of MNFS $\psi_t(x)$	
r	$\underline{\psi_a}(x)$	$\overline{\psi_a}(x)$	$\underline{\psi_t}(x)$	$\overline{\psi_t}(x)$
0	-0.666666666666667	0.666666666666667	-0.666666666664398	0.666666666665309
0.1	-0.600000000000000	0.600000000000000	-0.600005429870000	0.600059771260000
0.2	-0.533333333333333	0.533333333333333	-0.533333308543298	0.533333333387653
0.3	-0.466666666666667	0.466666666666667	-0.466666664432997	0.466666007599225
0.4	-0.400000000000000	0.400000000000000	-0.400043298755430	0.400005346899600
0.5	-0.333333333333333	0.333333333333333	-0.333338644320987	0.333333333995438
0.6	-0.266666666666667	0.266666666666667	-0.266666666666434	0.266666655433998
0.7	-0.200000000000000	0.200000000000000	-0.200554323567800	0.200000443399765
0.8	-0.133333333333333	0.133333333333333	-0.133333333775446	0.133333333338876
0.9	-0.066666666666667	0.066666666666667	-0.066666666666535	0.066666667765553
1	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000

Table 6 - Accuracy of solutions of example 6.2, ε=0.5,x=0.5.

The error $[E(x)]_r = \lvert[\psi_a(x)]_r - [\psi_t(x)]_r\rvert$	
$[\underline{E(x)}]_r$	$[\overline{E(x)}]_r$
2.26863E-12	1.35758E-12
5.42987E-06	5.97713E-05
2.479E-08	5.43197E-11
2.23367E-09	6.59067E-07
4.32988E-05	5.3469E-06
5.31099E-06	6.62105E-10
2.32647E-13	1.12327E-08
0.000554324	4.434E-07
4.42113E-10	5.54271E-12
1.31659E-13	1.09889E-09
0	0

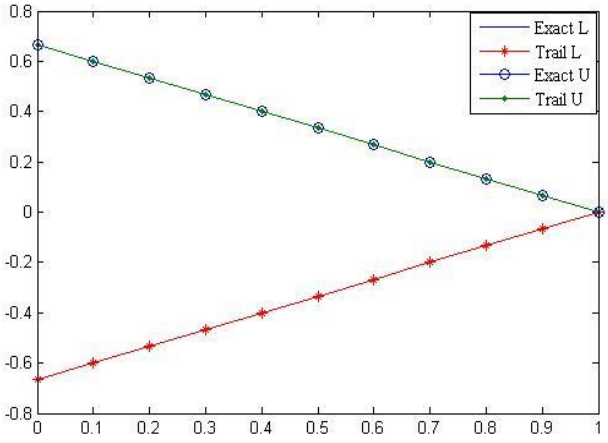

Fig. 5 - Analytic and neural solution of example 6.2, with ε=0.5.

References

[1] F. Marini," Comprehensive Chemometrics", 3.14 Neural Networks, Elsevier, 2009,https://doi.org/10.1016/B978-044452701-1.00128-9
[2] K.V. Shihabudheen,G.N. Pillai," Knowledge-Based Systems ", Elsevier 15 July 2018,https://doi.org/10.1016/j.knosys.2018.04.014
[3] Ali M. H.,"Design Fast Feed-Forward Neural Networks to Solve Two Point Boundary Value Problems", M.Sc. Thesis, College of Education Ibn Al-Haitham,University of Baghdad, Iraq, 2012.
[4] Hussein A. A. T., "Design Fast Feed-Forward Neural Networks to Solve Singular Boundary Value Problems", M.Sc. Thesis, College of Education Ibn Al-Haitham,University of Baghdad, Iraq, 2013.
[5] Tawfiq L. N. M., Al-Abrahemee K. M. M.,"Design Neural Network to Solve Singular Perturbation Problems", Applied and Computational Mathematics, Vol. 3, No. 3, 1-5, 2014 .
[6] 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
[7] 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.
[8] Zadeh, L. A., “Fuzzy Sets”, In Fuzzy Sets and Applications: Selected papers by L. A. Zadeh, Edited by Yager R. R., Ovehinnikos S., Tong R.M. and Ngnyen W.T., John Wiely and Sons, Inc., 1987.
[9] Dubois, D. and Prade, H., “Fuzzy Sets and Systems; Theory and Applications”, Academic Press, Inc., 1980.
[10] Yan, J., Ryan, M. and Power, J., “Using Fuzzy Logic: Towards Intelligent Systems”, Prentice Hall, Inc., 1994.
[11] Al-Doury, N. E., “Application of Fuzzy Set Theory in Character Cursive Recognition”, M.Sc. Thesis, College of Education, Ibn Al-Haitham, University of Baghdad, 2002.
[12] S. Effati, M. Pakdaman, "Artificial Neural Network Approach for Solving Fuzzy Differential Equations", Information Sciences, 180, 1434-1457, 2010.