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 Intelligent System For Solving Fuzzy Singular Perturbed Mixed Volterra - Fredholm Integro- Partial Differential Equations

Zainab Raad Al-shamary [a] **, *Khalid Mindeel Mohammed Al-Abrahemee*** [b]

[a]*Department of mathematics, College of Education, University of Al-Qadisiyah, Iraq. Email: zainabraad066@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: 22/05/2025
Rrevised form: 18/06/2025
Accepted : 25/06/2025
Available online: 30/09/2025

Keywords:

Mixed Volterra - Fredholm integra-differential equations; perturbed problems; feed forward neural network; hyperbolic tangent function, fuzzy sets

ABSTRACT

Through this effort, we have devised a new method using a neuro-fuzzy system. The neuro-fuzzy system method (MNFS) is the name given to this revised technique. We created a numerical technique for using BC to solve partial differential equations (FSPPs). WA suggests an updated approach that uses neural networks to solve singular perturbation problems.

We changed each element in the training set to a second-degree polynomial. For the purpose of solving FSPMV-FIDEs, the modified neuro-fuzzy system (MNFS) is the name given to this altered methodology

The fuzzy neural network's parameters and hidden units' sigmoid function,were determined using the activation function for hyperbolic tangents, which has the formula Q(x) =Q(x) The proposed approach was compared with classical training algorithms and analysis methods.
According to our research, the suggested method is notable for its high accuracy, low error rate, and significantly faster speed compared to traditional methods. The recommended approach is illustrated with several instances

MSC..

https://doi.org/10.29304/jqcsm.2025.17.32398

1. Introduction

The study of fuzzy integral equations which has garnered more attention recently ,has advanced quickly, especially when it comes to fuzzy control.Fuzzy number and arithmetic operations were initially presented by Zadeh [1]. Volterra-Fredholm integro-differential equations are involved in many different fields of science and engineering: Oceanography, fluid mechanics, electromagnetic theory, finance mathematics, plasma physics, population dynamics, These include biological processes and artificial neural networks. Singularly perturbed problems are characterized by the fact that the coefficient of the highest-order term in the equation is a very small parameter ε. Their approximate solutions have been studied in many articles and books. Population dynamics, fluid dynamics, heat transport issues,
nanofluids, mathematical biology, neurobiology, viscoelasticity, and simultaneous control systems are among the mathematical models displayed here.. can be listed in many applications in fields [2-8]. The perturbation parameter ε in the equation produces unlimited derivatives in the solution. Appropriate numerical methods should be preferred to eliminate this

∗Corresponding author: Zainab Raad Al-shamary

Email addresses: *zainabraad066@gmail.com*

Communicated by 'sub etitor'

situation. The fact that the problem examined in this study has both singular perturbation and integro-differential equation properties makes it difficult to obtain an analytical solution. Many methods have been developed to find solutions to integral and integro-differential equations. Some of these methods used basis functions to express the analytic form of the solution to transform the original problem, usually into an algebraic equation system. The techniques that we discuss now are artificial neural networks originally inspired by the operatizing of the human brain combined with intelligence. These are the ingredients to make a great number of artificial neurons that can be processed in an aligned way , so any problems related to aspects of classification can be solved now by the architecture of any identified ANN. Based on its position on the network, there are three types of computational neurons: input, output and hidden [9-13]

Here introduce the paper, and put a nomenclature if necessary, in a box with the same font size as the rest of the paper. The paragraphs continue from here and are only separated by headings, subheadings, images and formulae. The section headings are arranged by numbers, bold and 11 pt. Here follows further instructions for authors.

2. Basic definitions

Definition 2.1 [8]: Let X any set of nonempty points (the universal set is referred to as). A fuzzy set, known as $\mathcal{A}$, X is distinguished through a membership function
$U_{\mathcal{A}} = X \rightarrow I$, where I a interval of closed units [0, 1], and the collection of points $\mathcal{A}$ can be used to write the fuzzy set.$\mathcal{A} = \{(x, U_{\mathcal{A}}(x)) | x \in X, 0 \leq U_{\mathcal{A}}(x) \leq 1\}$
Remark 1 [9]:
It is important to notice that two law $\mathcal{A} \cup \mathcal{A}^c = X$ and $\mathcal{A} \cap \mathcal{A}^c = \emptyset$ For the fuzzy sets, they are broken, since $\mathcal{A} \cup \mathcal{A}^c \neq X$ and $\mathcal{A} \cap \mathcal{A}^c \neq \emptyset$. Indeed, for allx $\in$ X, if $U_{\mathcal{A}}(x) = \tau, 0 < \tau < 1$, then:

$$U_{\mathcal{A} \cup \mathcal{A}^c}(x) = \max\{\tau, 1 - \tau\} \neq 1$$
$$U_{\mathcal{A} \cap \mathcal{A}^c}(x) = \min\{\tau, 1 - \tau\} \neq 0$$

The r-Level sets

This section's objective is to discuss the essential and important features of what is known as the τ -level set,, which play an important role in fuzzy sets. There are further ways to connect fuzzy sets to conventional sets. τ -level sets are the collection that acts as a bridge between regular sets and fuzzy sets, and they can used to show that some results that satisfy in fuzzy sets also satisfy in regular sets.

The Proposed Method

This new strategy is predicated on substituting.Every x (input vector) in the training set
$\vec{x} = (x_1 , x_2 , \dots , x_n)$, $x_j \in [a, b]$ by a polynomial of second degree, it will appear as follows : $H(x) = \delta(\overrightarrow{x^2} + \vec{x} + 1)$, $\delta \in (a , b)$
Afterward, the input vector is going to be:
$(H(x_1) , H(x_2) , \dots , H(x_n))$,$H(x_j) \in (a , b)$ and $j = 1,2, \dots , n$.
Using M_{NFS}
Itis not advisable to instruct the neural networkwithin the range of the starting and finishing locations.; instead, trai ning points should be chosen throughout theopen period(a, b). Consequently, there is less computation error involv ed in the amount of calculations.In,fact,
The training points are converted into comparable points within the open interval determined by the selected distan ce [a, b] for neural network training
(a,b) In these connected fields, the network is trained by employing the novel technique.. ,we have the following from above: For a specified input vector $(x_1 , x_2 , \dots , x_n)$, $x_j \in [a, b]$ and $j = 1,2, \dots , n$.
the M_{NFS}'s output is: $[\xi]_\tau = \sum_{i=1}^{m} [\mathcal{v}_{kj}]_\tau \, Q([\aleph_{etij}]_\tau)$ For $i = 1, \dots , m$ is the entire quantity of units that are hidden where
$[\aleph_{etij}]_\tau = \sum_{j=1}^{n} [\mathcal{w}_{ji}]_\tau H(x_j) + [\mathcal{b}_j]_\tau$ and $H(x) = \delta(\overrightarrow{x^2} + \vec{x} + 1)$, $\delta \in (0 ,1)$
where $x_j \in [a , b]$ and $H(x_j) \in (a , b)$, $j = 1,2, \dots , n$
Note: For M_{NFS}:

$$\frac{d[\xi]_\tau}{d[\mathcal{v}_{kj}]_\tau} = \sum_{i=1}^{m} Q\left(\sum_{j=1}^{n} [\mathcal{w}_{ji}]_\tau H(x_j) + [\mathcal{b}_j]_\tau\right) = \sum_{i=1}^{m} Q\left(\sum_{j=1}^{n} \delta(\overrightarrow{x_j^2} + \vec{x_j} + 1) [\mathcal{w}_{ji}]_\tau + [\mathcal{b}_j]_\tau\right)$$

$$\frac{d[\xi]_\tau}{d[\mathcal{b}_j]_\tau} = \sum_{j=1}^{n} [\mathcal{v}_{kj}]_\tau Q'\left(\sum_{j=1}^{n} [\mathcal{w}_{ji}]_\tau H(x_j) + [\mathcal{b}_j]_\tau\right) = \sum_{i=1}^{m} Q'\left(\sum_{j=1}^{n} \delta(\overrightarrow{x^2} + \vec{x} + 1) [\mathcal{w}_{ji}]_\tau + [\mathcal{b}_j]_\tau\right)$$

$$\frac{d[\xi]_\tau}{d[\mathcal{w}_{ji}]_\tau} = \sum_{j=1}^{n} Q'\left(\sum_{j=1}^{n} [\mathcal{w}_{ji}]_\tau H(x_j) + [\mathcal{b}_j]_\tau\right) = Q'\left(\sum_{j=1}^{n} \delta(\overrightarrow{x^2} + \vec{x} + 1) [\mathcal{w}_{ji}]_\tau + [\mathcal{b}_j]_\tau\right)$$

where Q' is the activation function's first derivative

3. Solution of FSPMV-FIDE for PDE with B.C by M_{NFS}

Consider Consider two dimintion of FSPMV-FIDE for PDE

$$\varepsilon\left(\frac{\partial^n \xi(x,y)}{\partial x^n} + \frac{\partial^{n+m}\xi(x,y)}{\partial x^n\,\partial y^m} + \frac{\partial^m \xi(x,y)}{\partial y^m}\right) = \mathcal{M}(\xi, x, y, \varepsilon) + \int_0^x \int_\Omega K(x, y, t, r,),$$

$$G(\xi(t, r))\, dt dr \quad (x, y) \in [0, T] \times \Omega \qquad (1)$$

$\Omega = [a, b]$, $x, y \in I = [0, x].[0, y]$, $0 \le r \le (k - 1)$

where $\xi\,(x, y)$ is an unknown function that has to be identified, analytical functions are $\mathcal{M}(\xi, x, y)$ and $K(x, y, t, r,)$ on $D = [0, T] \times \Omega$ and $D \times \Omega^2$, Ω is a closed- bounded region, respectively.

Considering that ξ is a fuzzy function and that $\acute{\xi}$ its the fuzzy derivative, there might only be a hazy answer tonthis equation.

Let $\xi\,(x, y) = [[\xi^{lower}(x, y, \tau)], [\xi^{upper}(x, y, \tau)]]$ be first order fuzzy solution of FSPMV-FIDE , have the equivalent system , the dirichiet fuzzy BCs for x , $y \in [0,1]$

$\xi\,(0, y) = U_0(y), \xi\,(1, y) = U_1(y), \xi\,(x, 0) = V_0(x)\, and\, \xi\,(x, 1) = V_1(x)$

$U_0(y)$, $U_1(y)$, $V_0(x) and V_1(x)$ are fuzzy number or fuzzy function with

τ –level set:

$[U_0(y)]_\tau = [[U_0(y)]_\tau^{lower}, U_0(y)]_\tau^{upper}]$, $[U_1(y)]_\tau = [[U_1(y)]_\tau^{lower}, U_1(y)]_\tau^{upper}]$

$[V_0(x)]_\tau = [[V_0(x)]_\tau^{lower}, [V_0(x)]_\tau^{upper}]]$ $[V_1(x)]_\tau = [\,[V_1(x)]_\tau^{lower}, [V_1(x)]_\tau^{upper}]$

The fuzzy trial solution

$$[\xi_t(x, y)]_\tau = [[\xi_t(x, y)]_\tau^{lower}, [\xi_t(x, y)]_\tau^{upper}] \qquad (2)$$

$$[\xi_t(x, y)]_\tau^{lower} = [J(x, y)]_\tau^{lower} + (1-x)(1-y)(\,\mathcal{O}_{tk}(H(x), H(y), H(\tau),$$

$$\rho_\tau^{lower}, \varepsilon)]_\tau^{lower} \qquad (3)$$

$$[\xi_t(x, y)]_\tau^{upper} = [J(x, y)]_\tau^{upper} + (1-x)(1-y)(\,\mathcal{O}_{tk}(H(x), H(y), H(\tau),$$

$$\rho_\tau^{upper}, \varepsilon)]_\tau^{upper} \qquad (4)$$

Where $[J(x, y)]_\tau^{lower}$ *and* $[J(x, y)]_\tau^{upper}$ are selected to meet the fuzzy BCs, namely

$$J(x, y)]_\tau^{lower} = (1 - x)\, U_0(y)]_\tau^{lower} + xU_1(y)]_\tau^{lower} + (1 - y)[[V_0(x)]_\tau^{lower} - [(1 - x)\, V_0(0)]_\tau^{lower} + xV(1) + y\,[V_0(x)]_\tau^{lower} - [(1 - x)\, V_0(0)]_\tau^{lower} + x$$

$$V_0(1)]_\tau^{lower}]] \qquad (5)$$

$$J(x, y)]_\tau^{upper} = (1 - x)U_0(y)]_\tau^{upper} + xU_1(y)]_\tau^{upper} + (1 - y)\,[[V_0(x)]_\tau^{upper} - [(1 - x)\, V_0(0)]_\tau^{upper} + xV(1) + y\,[V_0(x)]_\tau^{upper} - [(1 - x)\, V_0(0)]_\tau^{upper} + x$$

$$V_0(1)]_\tau^{upper}] \qquad (6)$$

The minimized error function will be :

$$\mathbb{E}(\rho) = \sum_{i=1}^{g} [\mathbb{E}(\rho)_\tau^{lower}], [\,\mathbb{E}(\rho)_\tau^{upper}]),\ \rho \text{ isparimeter} \qquad (7)$$

$\mathbb{E}(\rho,\varepsilon)_{i\tau}^{lower}=[[\ \varepsilon(\frac{\partial^{n}\xi(x,y)}{\partial x^{n}}+\frac{\partial^{n+m}\xi(x,y)}{\partial x^{n}\,\partial y^{m}}+\frac{\partial^{m}\xi(x,y)}{\partial y^{m}})]_{\tau}^{lower}-\frac{1}{\varepsilon}(\ [\sum_{x_I,y_i\in D}\mathcal{M}($

$\xi\,,H(x_i),H(y_i),H(\tau),\varepsilon)\ ]_{\tau}^{lower}+[\int_0^x\int_\Omega\ K(H(x_i),H(y_i),H(\tau),t,r)\ G(\xi(t,r)$

$dt dr]_{\tau}^{lower}\ ]^2$ (8)

$\mathbb{E}(\rho,\varepsilon)_{i\tau}^{upper}=[[\ \varepsilon(\frac{\partial^{n}\xi(x,y)}{\partial x^{n}}+\frac{\partial^{n+m}\xi(x,y)}{\partial x^{n}\,\partial y^{m}}+\frac{\partial^{m}\xi(x,y)}{\partial y^{m}})]_{\tau}^{upper}-\frac{1}{\varepsilon}(\ [\sum_{x_I,y_i\in D}\mathcal{M}($

$\xi\,,H(x_i),H(y_i),H(\tau),\varepsilon)\ ]_{\tau}^{upper}+[\int_0^x\int_\Omega\ K(H(x_i),H(y_i),H(\tau),t,r)\ G(\xi(t,r)$

$dt dr]_{\tau}^{upper}\ ]^2$ (9)

4. Numerical results

We give a lot of examples to show the behavior and effectiveness of the proposed (M_{NFS}), and MATLAB version 7.12 is used to develop the applications .For ODE's we suggest a three-layer feed- forward neural network (NN) with one input unit and seven hidden units in one hidden layer and a linear output unit ,but use 9 hidden units in PDE's . The activation A hyperbolic tangent function function serves as the hidden units activation function, and its expression : $Q(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$.

Example1.4 : Consider the second-order linear PMV-FIDE:

$$\varepsilon\left(\frac{\partial^2\xi(x,y)}{\partial x^2}+\xi(x,y)\right)=\mathcal{M}(x,y,\tau)+\int_0^x\int_0^1 t e^{s}\xi(s,t)dt\,ds$$

$t, x\in[0,1],\ x,y\in[0,x].[0,y]\ ,\ 0\le x,y\le 1,\ \ 0<\varepsilon<<1\ ,\tau\in[0,1]$

Where

$\mathcal{M}(x,y,\tau)=-\frac{1}{12}+\frac{1}{3}e^{x}-\frac{1}{4}e^{x}\cos(x)+\frac{1}{4}e^{x}\sin(x)+\sin(x)+y\,[(5+\tau),(5-\tau))]$

with fuzzy B.C

$\xi(0,y)=[y((5+\tau),y(5-\tau))]$,

$\xi(1,y)=[\sin(1)+y(5+\tau),\sin(1)+y\,(5-\tau)]$,

$\xi(x,0)=[\sin(x)\,,\sin(x)]$,

$\xi(x,1)=[\sin(x)+(5+\tau),\sin(x)+(5-\tau)]$

Therefore, the following is a fuzzy trial solution for this example is :

$[\xi_t(x,y]_{\tau}^{lower}=[\ (1\ -\ x)(y(5+\tau)+x(\sin(1)+y(5+\tau))+(1-y)$

$[\sin(x-[(1\ -\ x)\sin(0\ \ +y\,[(\sin(x)+(5+\tau)-[(1\ -\ x)\sin(0+x((\sin x\ +(5+\tau))+xy(1-x)(1\ -$ $y)\ [\mathcal{O}_{tk}(H(x),H(y),H(\tau),\rho,\varepsilon)]_{\tau}^{lower}$

$[\xi_t(x,y]_{\tau}^{upper}=[\ (1-x)(y(5-\tau).x(\sin(1)+y(5-\tau)+(1-y)\ [\sin(x)-[(1-x)\sin(0)+x\,V(1)\ \ +y\,[(\sin(x)+$ $(5-\tau)-[(1-x)\sin(0)+x(\sin x\ +(5-\tau)\ ]+xy(1-x)(1-\ y)\ [\mathcal{O}_{tk}(H(x),,H(y)\ ,H(\tau),\rho,\varepsilon]_{\tau}^{upper}$

the input vector $\vec{x},\vec{y}$ (training set) is :

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

$\vec{y}=(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)$, and The M_{NFS} was trained using a grid of ten equally spaced points in the interval (0,1) (i.e)

We now use the following procedures to determine the error function $\mathbb{E}$ that needs to be minimized for this situation.

$\frac{d[\xi_t(xi,\rho)\ ]_{\tau}^{lower}}{dx}=[-(y(5+\tau)+(\sin(1)+y(5+\tau)+(1-y)[\ \cos(x)$

$-[-\sin(0)+\sin(1)\ \ +y\,[(\cos(x)-[-\sin(0)+\sin(1)]]+xy(1-x)(1\ -\ y)\frac{d[\mathcal{O}_{tk}(H(x),H(y),H(\tau),\rho,\varepsilon)\ ]_{\tau}^{lower}}{dx}+y(1\ -$ $y)\ [\mathcal{O}_{tk}(H(x),H(y),H(\tau),\rho,\varepsilon)\ ]_{\tau}^{lower}$

$\frac{d[\xi_t(xi,\rho)\ ]_{\tau}^{upper}}{dx}=[-(y(5+\tau)+(\sin(1)+y(5+\tau)+(1-y)[\ \cos(x)$

$-[-\sin(0)+\sin(1)\ \ +y\,[(\cos(x)-[-\sin(0)+\sin(1)]]+xy(1-x)(1\ -\ y)\frac{d[\mathcal{O}_{tk}(H(x),H(y),H(\tau),\rho,\varepsilon)\ ]_{\tau}^{upper}}{dx}+y(1\ -$ $y)\ [\mathcal{O}_{tk}(H(x),H(y),H(\tau),\rho,\varepsilon)\ ]_{\tau}^{upper}$

$\frac{d^2[\xi_t(x)]_\tau^{lower}}{dx^2} = [-(1-y)\sin x + (-y\sin(x))\, xy(1-x)(1-y)$
$\frac{d^2[[\mathcal{O}_{tk}(H(x),(H(\tau),\rho,\varepsilon)]_\tau^{lower}}{dx^2} + y(1-y)\ \frac{d[\mathcal{O}_{tk}(H(x),H(y),H(\tau),\rho,\varepsilon)]_\tau^{lower}}{dx}$
$\frac{d^2[\xi_t(x)]_\tau^{upper}}{dx^2} = [-(1-y)\sin x + (-y\sin(x))\, xy(1-x)(1-y)$
$\frac{d^2[[\mathcal{O}_{tk}(H(x),(H(\tau),\rho,\varepsilon)]_\tau^{upper}}{dx^2} + y(1-y)\ \frac{d[\mathcal{O}_{tk}(H(x),H(y),H(\tau),\rho,\varepsilon)]_\tau^{upper}}{dx}$

The error function that needs to be reduced for this issue is : $\mathbb{E}_{i\tau}^{lower}(\rho,\varepsilon)=[[-(1-y)\sin x + (-y\sin(x))$ $xy(1-x)(1-y)$
$\frac{d^2[[\mathcal{O}_{tk}(H(x),(H(\tau),\rho,\varepsilon)]_\tau^{lower}}{dx^2} + y(1-y)\ \frac{d[\mathcal{O}_{tk}(H(x),H(y),H(\tau),\rho,\varepsilon)]_\tau^{lower}}{dx}$
$]_\tau^{lower} - \frac{1}{\varepsilon}[\sum_{j=1}^{7}(-\frac{1}{12}+\frac{1}{3}e^x-\frac{1}{4}e^x\cos(x)+\frac{1}{4}e^x\sin(x)+\sin(x)+y\,[(5+\tau)]+\int_0^x\int_0^1 t e^s\xi_t(s,t)dt\,ds]_\tau^{lower}]^2$
$\mathbb{E}_{i\tau}^{upper}(\rho,\varepsilon)=[[-(1-y)\sin x + (-y\sin(x))\, xy(1-x)(1-y)$
$\frac{d^2[[\mathcal{O}_{tk}(H(x),(H(\tau),\rho,\varepsilon)]_\tau^{lower}}{dx^2} + y(1-y)\ \frac{d[\mathcal{O}_{tk}(H(x),H(y),H(\tau),\rho,\varepsilon)]_\tau^{lower}}{dx}$
$]_\tau^{lower} - \frac{1}{\varepsilon}[\sum_{j=1}^{7}(-\frac{1}{12}+\frac{1}{3}e^x-\frac{1}{4}e^x\cos(x)+\frac{1}{4}e^x\sin(x)+\sin(x)+y\,[(5+\tau)]+\int_0^x\int_0^1 t e^s\xi_t(s,t)dt\,ds]_\tau^{lower}]^2$

Since $[\mathcal{O}_{tk}(H(x),(H(\tau),\rho,\varepsilon)]_\tau^{lower} = \sum_{j=1}^{9}[\,v_{kj}]_\tau^{lower}\,Q([w_{ji}]_\tau^{lower}H(x_j)+[b_j]_\tau^{lower})$
$[\mathcal{O}_{tk}(H(x),H(\tau),\rho,\varepsilon)]_\tau^{upper} = \sum_{j=1}^{9}[\,v_{kj}]_\tau^{upper}\,Q([w_{ji}]_\tau^{upper}H(x_j)+[b_j]_\tau^{upper})$
$\frac{d[[\mathcal{O}_{tk}(H(x),(H(\tau),\rho,\varepsilon)]_\tau^{lower}}{dx} = \sum_{i=1}^{9}\delta[\,v_{kj}]_\tau^{lower}[w_{ji}]_\tau^{lower}\,Q'((H(x_j)[w_{ji}]_\tau^{lower}+[b_j]_\tau^{lower})$
$\frac{d[[\mathcal{O}_{tk}(H(x),(H(\tau),\rho,\varepsilon)]_\tau^{upper}}{dx} = \sum_{i=1}^{9}\delta[\,v_{kj}]_\tau^{upper}[w_{ji}]_\tau^{upper}\,Q'((H(x_j)[w_{ji}]_\tau^{upper}+[b_j]_\tau^{upper})$

And so :
$\frac{d^2[[\mathcal{O}_{tk}(H(x),(H(\tau),\rho,\varepsilon)]_\tau^{lower}}{dx^2} = \sum_{i=1}^{9}(\delta^2[\,v_{kj}]_\tau^{lower}\left([w_{ji}]_\tau^{lower}\right)^2 Q''((H(x_j)$
$[w_{ji}]_\tau^{lower} + [b_j]_\tau^{lower}))$
$\frac{d^2[[\mathcal{O}_{tk}(H(x),(H(\tau),\rho,\varepsilon)]_\tau^{upper}}{dx^2} = \sum_{j=1}^{9}(\delta^2[\,v_{kj}]_\tau^{upper}\left([w_{ji}]_\tau^{upper}\right)^2 Q''((H(x_j)$
$[w_{ji}]_\tau^{upper}+[b_j]_\tau^{upper}))$

Also $Q = 2Q^3 - 2Q$.

$\mathbb{E}(\rho)_{i\tau}^{lower}=[[[-(1-y)\sin x + (-y\sin(x)) + xy(1-x)(1-y)$
$\sum_{i=1}^{9}(\delta^2[\,v_{kj}]_\tau^{lower}\left([w_{ji}]_\tau^{lower}\right)^2(2Q^3-2Q)((H(x_j)[w_{ji}]_\tau^{lower}+[b_j]_\tau^{lower}))$
$+\ y(1-y)\sum_{i=1}^{9}\delta[\,v_{kj}]_\tau^{lower}[w_{ji}]_\tau^{lower}\,Q'((H(x_j)[w_{ji}]_\tau^{lower}+[b_j]_\tau^{lower})]_\tau^{lower} - \frac{1}{\varepsilon}[\sum_{j=1}^{7}(-\frac{1}{12}+\frac{1}{3}e^x-\frac{1}{4}e^x\cos(x)+$
$\frac{1}{4}e^x\sin(x)+\sin(x)+y\,[(5+\tau)]+\int_0^x\int_0^1 t e^s\xi_t(s,t)dt\,ds]_\tau^{lower}]^2$

$\mathbb{E}(\rho)_{i\tau}^{upper}=[[-(1-y)\sin x + (-y\sin(x)) + xy(1-x)(1-y)$
$\sum_{i=1}^{9}(\delta^2[\,v_{kj}]_\tau^{upper}\left([w_{ji}]_\tau^{upper}\right)^2(2Q^3-2Q)\,((H(x_j)[w_{ji}]_\tau^{upper}+[b_j]_\tau^{upper}))$
$+y\,(1-y)\sum_{i=1}^{9}\delta[\,v_{kj}]_\tau^{upper}[w_{ji}]_\tau^{upper}\,Q'((H(x_j)[w_{ji}]_\tau^{upper}$
$+[b_j]_\tau^{upper})]_\tau^{upper} - \frac{1}{\varepsilon}[\sum_{j=1}^{7}(-\frac{1}{12}+\frac{1}{3}e^x-\frac{1}{4}e^x\cos(x)+\frac{1}{4}e^x\sin(x)+\sin(x)+y\,[(5+\tau)]+\int_0^x\int_0^1 t e^s\xi_t(s,t)dt\,ds]_\tau^{upper}]^2$

Since x in this example is between 0 and 1,which requires to select $0< \delta < 0.3$, and let ε= 10-3 .
In this training set will be $\vec{x} = (0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)$,
H(x) : (0.3, 0.333, 0.372, 0.417, 0.468, 0.525, 0.588, 0.657, 0.732, 0.813, 0.9)
$\vec{y} = (0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)$,
H(y) : (0.3, 0.333, 0.372, 0.417, 0.468, 0.525, 0.588, 0.657, 0.732, 0.813, 0.9)

Assessing the performance gradient in connection to the coefficient is simple. the analytic and neutrally netural solution in the training set, which was used to train the feedforward NN is displayed in an equal equidistant grid of points at [0,1] in figure (1). The exact response and train's accuracy errors are then provided by the oral result of M_{NFS}, tables (1.1) and (1.2) .

Analytic and FFNN solution for example(1)are shown in table (1.1) where ε= 10-3, τ = 0.8

input	input	"Analytic solution"		"Solution of FFNN $\xi_t(x)$"	
x	y	$[\xi_a(x)]_\tau^{lower}$	$[\xi_a(x)]_\tau^{upper}$	$[\xi_t(x)]_\tau^{upper}$	$[\xi_t(x)]_\tau^{upper}$
0	0	0	0	0	0
0.1	0.1	0.679833417	0.519833417	0.679833417	0.519833417
0.2	0.2	1.358669331	1.038669331	1.358669331	1.038669331

0.3	0.3	2.035520207	1.555520207	2.035520207	1.555520207
0.4	0.4	2.709418342	2.069418342	2.709418342	2.069418343
0.5	0.5	3.379425539	2.579425539	3.379425567	2.579425511
0.6	0.6	4.044642473	3.084642473	4.044642473	3.084642474
0.7	0.7	4.704217687	3.584217687	4.704217687	3.584217687
0.8	0.8	5.357356091	4.077356091	5.357356091	4.0773561
0.9	0.9	6.00332691	4.56332691	6.00332691	4.56332691
1	1	6.641470985	5.041470985	6.64147098	5.041470988

Correctness of the example(1)'s solution in table(1.2) where ε= 10-3, $\tau = 0.8$

The error $[\mathbb{E}(x)]_\tau = \left|[\xi_a(x)]_\gamma - \xi_t(x)]_\tau\right|$	
$[\text{error}]_\tau^{\text{lower}}$	$[\text{error}]_\tau^{\text{upper}}$
0	0
4.86056E-13	1.50998E-11
5.0681E-11	7.19711E-11
3.96998E-11	2.2598E-10
8.94698E-11	4.16459E-10
2.79278E-08	2.73058E-08
8.44658E-13	3.80485E-10
5.58096E-11	2.84608E-11
3.29932E-11	8.9264E-09
2.54969E-11	3.63167E-11
4.7848E-09	2.9151E-09

Correctness of the example(1)'s solution in table(1.3) where ε= 10-3, $\tau = 0.8$

The error $[\mathbb{E}(x)]_\tau = \left|[\xi_a(x)]_\gamma - \xi_t(x)]_\tau\right|$	
$[\text{error}]_\tau^{\text{lower}}$	$[\text{error}]_\tau^{\text{upper}}$
0	0
1.9524E-09	4.21E-08
4.40131E-06	1.37E-06
7.23921E-07	3.83E-08
2.93139E-08	2.08E-07
1.27325E-07	5.55E-07
3.34852E-09	4.61E-05
2.50821E-09	1.35E-07
6.25166E-08	3.48E-10
8.96736E-08	6.39E-08
2.39631E-12	3.14E-11
MSE=1.99246E-11	MSE=2.12E-09

Note : By resolving the above example without updating, we found that $[\text{error}]_{\tau}^{\text{lower}}$ =8.02873E-16 and $[\text{error}]_{\tau}^{\text{upper}}$ =8.34162E-16 and therefore we notice that the updated method is better.

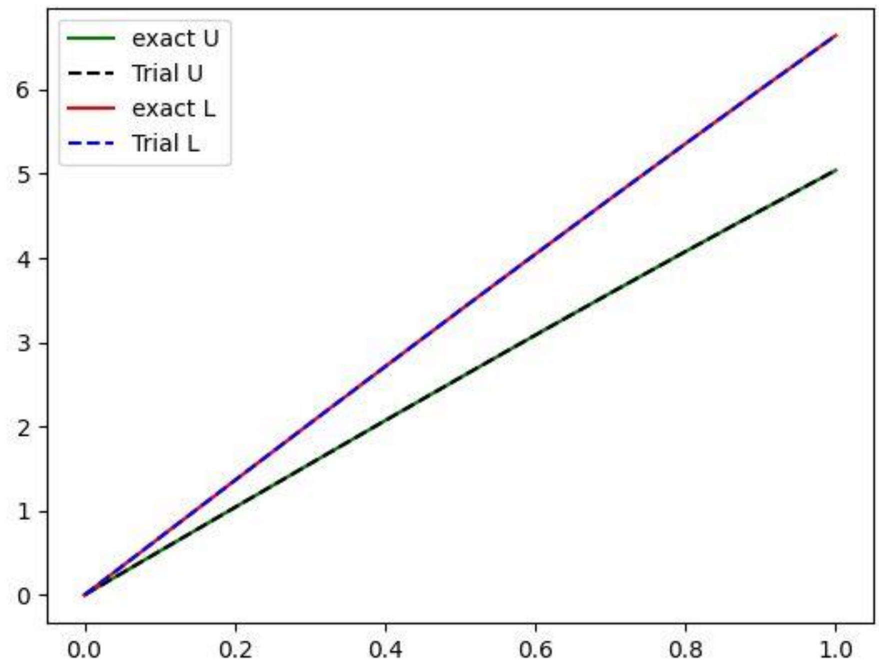

Figure 1: Neural and analytical solution for example (1), with ε =10-3

REFERENCES

[1]. S. Chang and L. Zadeh, "On fuzzy mapping and control," IEEE Trans. Systems, Man cybernet, (1972), 2, pp.30–34.

[2] . Al-Abrahemee, K.M.M.,Efficient Algorithm for Solving Fuzzy Singularly Perturbed Volterra Integro-Differential Equation,Iraqi Journal of Science., 64(11), pp. 5851–5865,2023. https://orcid.org/0000-0002-4705-6349.

[3]. 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/10.1080/09720502.2021.2001136.

[4]. Rusul Kareem, K.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.

[5]. R.F. Kadam, K.M.M. Al-Abrahemee, On convergence of the Levenberg-Marquardt method underlocal error bound, problems, Journal of Interdisciplinary Mathematics, 25(5), 1495-1508, (2022), https://doi.org/10.1080/09720502.2022.2079228.

[6]. Al-Abrahemee, K.M., Jabber, A.K., High Performance Training Algorithm with Strong Local Convergence Properties for Solving Some Fractional Integral Equation, AIP Conference Proceedings, Volume 2414, Issue 1,2023. https://doi.org/10.1063/5.0115034.

[7]. S. Haykin , " Neural networks: A comprehensive foundation",(1993), https://dl.acm.org/doi/10.5555/541500.

[8].R.M.Hristev"The ANN Book ", Edition 1, (1998), https://www.academia.edu/44105267/Hritsev_The_ANN_Book

[9]. L. Hooshangian., "Nonlinear Fuzzy Volterra Integro-diferential Equation of N-th Order: Analytic Solution and Existence and Uniqueness of Solution," Int. J. Industrial Mathematics, vol. 11, no. 1, (2019.) https://www.researchgate.net/publication/329894133.

[10]. R.F. Kadam, K.M.M. Al-Abrahemee. Neuro-fuzzy system for solving fuzzy singular perturbation problems. Journal of Interdisciplinary Mathematics. 25(5), 1509-1524, 2022. https://doi.org/10.1080/09720502.2022.2079229

[11]. Al-Abrahemee, K.M.M.. A novel hybridized neuro-fuzzy model for solving fuzzy singular perturbation problems with initial conditions,Journal of Interdisciplinary Mathematics, 26(6), pp. 1287–1301,2023, https://doi.org/10.47974/JIM-1627

[12]. D. and Kruse, R. Nauck, "Neuro-Fuzzy Classification with NEFCLASS," in Operations Research Proceedings, Berlin, pp. pp. 294-299,(1996). ISBN : 978-3-540-60806-6.

[13]. Tabark Aqeel Al-Janabi*, Khalid Mindeel Mohammed Al-Abrahemee., " Solving Singular Perturbation Problems With Initial and Boundary Conditions By Using Modified Neuro System," Iraqi Journal of Science, 2024, Vol. 65, No. 11, pp: 6566- 6575.(2024). https:// DOI: 10.24996/ijs.2024.65.11.31.