Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44)** : 1–8.
ISSN-0037-8712
doi:10.5269/bspm.82204

Numerical Simulation for Fuzzy Perturbed Integro-Differential Equations Using partially Neuro-Fuzzy System

Thaer Ubaid Al-Jubouri and Khalid Mindeel Mohammed Al-Abrahemee

ABSTRACT: Recently, the study of fuzzy singular perturbed Fredholm integro- differential equations (FSP-FIDEs) has been of increasing interest for a long time. Our paper has a design for a fast feed -forward partially neuro-fuzzy system (PFNN) to adduce a new method for solving two- dimensions (FSPFIDEs). Employing a multi-layer that has one hidden layer consisting of seven a set of units and one linear output node. And the sigmoid activation for every unit is the hyperbolic tangent function and the Levenberg – Marquardt training algorithm(LM) . We compared our exact solution in illustrative examples with the results of numerical experiments, confirming the efficiency and accuracy of our presented scheme.

Keywords: Fredholm integro-differential equations, perturbed problems, feed forward neural network, hyperbolic tangent function, fuzzy sets.

Contents

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]. 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, artificial neural networks and biological processes are among these fields. 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. The mathematical models seen here are population dynamics, fluid dynamics, heat transport problem, nanofluid, neurobiology, mathematical biology, viscoelasticity and simultaneous control systems etc. can be listed in many applications in fields [2,3,4,5,6,7,8]. The perturbation parameter ε in the equation produces unlimited derivatives in the solution. Appropriate numerical methods should be preferred to eliminate this 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 multiple hidden units with a single linear output neuron. These are the ingredients to make a great

2020 *Mathematics Subject Classification:* 34K28.
Submitted February 25, 2026. Published June 02, 2026.

Typeset by $\mathcal{B}^{S}\mathcal{P}_{\mathcal{M}}$style.
© Soc. Paran. de Mat.

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 artifation neural network (ANN) . Based on its position on the network, there are three types of computational neurons: input, output and hidden [9,10,11,12,13].

In this paper, we contemplate the two-dimensional singular perturbed fuzzy Fredholm integro- differential equations.

$$\varepsilon\Phi^{(n)}(x,y) = \mathbf{F}(\Phi^{'}, \Phi^{2}, \Phi^{(n-1)}x, y, \varepsilon) + \int_a^b \int_c^d \mathbf{K}(x,y,t,s)\Phi(t,s)dtds$$

where ε is perturbation parameter $(0 < \varepsilon \ll 1)$ and $(x, y) \in [a, b] \times [c, d]$ With the fuzzy (IC): $\Phi(a, y) = A_1\Phi(a, y) = A_2 ... \Phi(a, y) = A_{n-1}$, where A_i are fuzzy number in E_1 with ω-level, the units are grouped into sets and connected to a single linear output unit, $[A_i]_\omega = [[A_i]^l_\omega, [A_i]^u_\omega]$. Or with the fuzzy (BC): $\Phi(a) = [A]_\omega, \Phi(b) = [B]_\omega$.

Where $\Phi(t, s)$ is an unknown function that must be identified and The kernel K and the data function F are assumed to be sufficiently smooth. Under suitable conditions on F and K, for any $\varepsilon > 0$.

2. Basic Definitions

Definition 2.1 *[8] Let X any set of nonempty points (referred to as the universal set). A fuzzy set, known as $\mathcal{A}$, X is distinguished by a membership function*
$U_{\mathcal{A}} = X \to I$, where I a closed unit interval $[0, 1]$, 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 2.1 *[9] It is important to notice that two law $\mathcal{A} \cup \mathcal{A}^c = X$ are broken for the fuzzy sets, since $\mathcal{A} \cup \mathcal{A}^c \neq X$ and $\mathcal{A} \cup \mathcal{A}^c \neq \phi$. Indeed, for all $x \in X$, if $U_{\mathcal{A}}(x) = r, 0 < \omega < 1$, then:*

$$U_{\mathcal{A}\cup\mathcal{A}^c}(x) = max\{r, 1 - r\} \neq 1$$

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

The r-Level Sets

The current section's purpose is to discuss the essential and key characteristics of the so-called r- contour sets, which play an important role in fuzzy sets. There are further ways to connect fuzzy sets to conventional sets. ω-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.

3. Architecture Partialy Neuro – Fuzzy System (PNFS)

In this study, we consider a fuzzy three-layer PFNN composed of n input units, m hidden units, and s output units. the target vector, the connection fuzzy weights, and the bias terms are real numbers, and the input vector consists of real-valued elements. The dimension of the PFNN is denoted by the number of neurons in each layer, namely $(n \times m \times s)$, where n, m, and s represent the numbers of neurons in the input, hidden, and output layers, respectively. The architecture of the model illustrates how the PFNN transforms the n input variables into the output vector. $(x1, x2, ...xi, ...xn)$ into the s fuzzy outputs: $([OUT_{k_1}]_\omega, [OUT_{k_2}]_\omega, [OUT_{k_3}]_\omega, ..., [OUT_{k_S}]_\omega)$ distributed across the hidden layer consisting of m fuzzy neurons $([Hid_1]_\omega, [Hid_2]_\omega, [Hid_3]_\omega, ..., [Hid_j]_\omega, ..., [Hid_m]_\omega)$ where $\mathfrak{b}_j$ and g_k are biases terms for the fuzzy neurons $[Hid_j]_\omega, [OUT_k]_\omega$ respectively, $[w_{ji}]\omega$ are the fuzzy connection weight between the crisp neuron xi and the fuzzy neuron$[Hid_j]_\omega$, and $[v_{kj}]\omega$ are the fuzzy synaptic weight between fuzzy neurons $[Hid_j]\omega$ to fuzzy neuron $[OUT_k]_\omega$.

Input unit

$x = x_i, i = 1, 2, 3, ..., n$

Hidden unit

$[Hid_j]_\omega = [[Hid_j]^l_\omega, [Hid_j]^u_\omega] = \Omega([net_j]_\omega) = [\Omega[net_j]]^l_\omega, \Omega[net_j]^u_\omega], j = 1, 2, 3, ..., m$

Where

$[net_j]^l_\omega = \sum_{i=1}^n x_i[w_{ji}]^l_\omega + \mathfrak{b}_j$

$[net_j]^u_\omega = \sum_{i=1}^{n} x_i [w_{ji}]^u_\omega + \mathfrak{b}_j$

Output unit

$[OUT_k]_\omega = [[OUT_k]^l_\omega], [[OUT_k]^u_\omega] = \Omega([net_{jk}]_\omega), k = 1, 2, 3, ..., s$

Where

$[net_j]^l_\omega = (\sum_{j\in a} [v_{kj}]^l_\omega [Hid_j]^l_\omega + \sum_{j\in b} [v_{kj}]^l_\omega [Hid_j]^u_\omega) + g_k$

$[net_j]^u_\omega = (\sum_{j\in c} [v_{kj}]^u_\omega [Hid_j]^u_\omega + \sum_{j\in d} [v_{kj}]^u_\omega [Hid_j]^l_\omega) + g_k$

such that $[Hid_j]^u_\omega \geq [Hid_j]^l_\omega \geq 0$ where

$a = \{j : [v_{kj}]^l_\omega \geq 0\}, b = \{j : [v_{kj}]^l_\omega < 0\},$

$c = \{j : [v_{kj}]^u_\omega \geq 0\}, d = \{j : [v_{kj}]^u_\omega < 0\}$

4. Solve FSPMF-FIDES for PDE with BC

To solve any FSPFIDEs for ODE by PFNN, we consider a three of NN layer containing a dual-unit network elements terms x and y, one concealed layer made up of m nonlinear activation functions and a single output neuron (x, y). in this work, the PFNN's $(2 \times m \times 1)$ dimension is displayed in figure (1). Additionally, each unit of our NN's input-output can be modified for the ω-level set in the manner described below:

Input unit

$x = x, y = y$

Hidden unit

$[Hid_j]_\omega = [[Hid_j]^l_\omega, [Hid_j]^u_\omega = [\Omega[net_j]^l_\omega, \Omega[net_j]^u_\omega]$

$[net_j]^l_\omega = x[w_{j1}]^l_\omega + y[w_{j2}]^l_\omega + \mathfrak{b}_j$

$[net_j]^u_\omega = x[w_{j1}]^l_\omega + y[w_{j2}]^u_\omega + \mathfrak{b}_j$

Output unit

$[OUT_k]_w = [[OUT_k]^l_\omega, [OUT_k]^u_\omega]$

$[OUT_k]^l_\omega = \sum_{j\in a}^{m} [v_j]^l_\omega [Hid_j]^l_\omega + \sum_{j\in b}^{m} [v_j]^l_\omega [Hid_j]^u_\omega$

$[OUT_k]^u_\omega = \sum_{j\in c}^{m} [v_j]^u_\omega [Hid_j]^u_\omega + \sum_{j\in d}^{m} [v_j]^u_\omega [Hid_j]^l_\omega$

for $[Hid_j]^u_\omega \geq [Hid_j]^l_\omega \geq 0$, where:

$a = \{j : [v_j]^l_\omega \geq 0\}, b = \{j : [v_j]^l_\omega < 0\},$

$c = \{j : [v_j]^u_\omega \geq 0\}, d = \{j : [v_j]^u_\omega < 0\}$

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

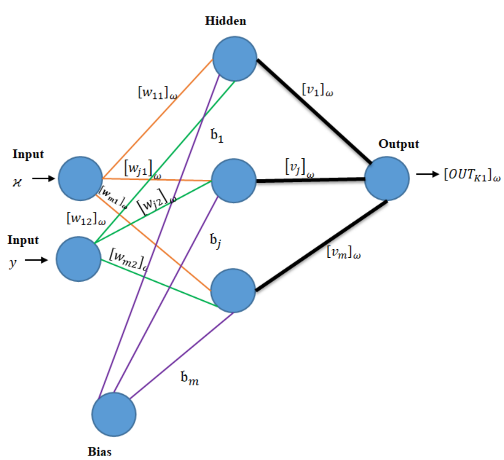

Figure 1: $(2 \times m \times 1)$ feed forward PNFS

5. Illustration of the Proposed Method

We consider a first-order two-dimensional problem FSPFIDEs for ODEs

$$\varepsilon\Phi^{'}(x,y) = F(\Phi,x,y,\varepsilon) + \int_a^b \int_c^d K(x,y,t,s)\Phi(t,s)dtds \tag{5.1}$$

, $(x,y) \in [a,b] \times [c,d]$ where ε denotes the perturbation parameter $(0 < \varepsilon \ll 1)$.
With the fuzzy (IC): $\Phi(a,y) = A$, where A is fuzzy number in $E1$ with ω-level.
Sets, $[A]_\omega = [[A]^l_\omega, [A]^u_\omega]$
Suppose that the kernel K and the target function F are smooth. Under appropriate conditions on K and F, for every $\varepsilon > 0$ .the equation (5.1) admits a unique continuous solution on $[a,b] \times [c,d]$ and This equation admits only fuzzy solutions.
Let $\Phi(x,y) = [[\Phi^l(x,y,\omega)], [\Phi^u(x,y,\omega)]]$ represent a fuzzy solution of a first-order two-dimensional system FSPFIDE, have the equivalent system
$\varepsilon[\Phi^{'}(x,y)]^l_\omega = [F(\Phi,x,y,\varepsilon)]^l_\omega + \big[\int_a^b \int_c^d K(x,y,t,s)\Phi(t,s)dtds\big]^l_\omega$
$\varepsilon[\Phi^{'}(x,y)]^u_\omega = [F(\Phi,x,y,\varepsilon)]^u_\omega + \big[\int_a^b \int_c^d K(x,y,t,s)\Phi(t,s)dtds\big]^u_\omega$
for $\omega \in [0,1]$. Suppose. $K(x,t)$ be continuous in $[0,1]$ and changes its sing in finite points for fix t.
For the given problem, the fuzzy trial solution is given by: $[\Phi_t(x,p)]_\omega = [A]_\omega + (x-a)[OUT(x,p,\varepsilon)]_\omega$
Where OUT(x,p,ε) is the feed-forward network output PFNS with one input unit for x and p so that The amount of error which is minimized is given by $\mathbb{E}(p) = \sum_{i=1}^{g}([\mathbb{E}(p)]^l_{i\omega} + [\mathbb{E}(p)]^u_{i\omega})$
where

$$[\mathbb{E}(p)]^l_{i\omega} = \Big[\Big[\frac{d\Phi_t(x_i,p)}{dx}\Big]^l_\omega - \frac{1}{\varepsilon}\Big[\sum_{x_i,y_i,\in D} F(x_i,y_i,\Phi_t(x_i,p),\varepsilon) + \int_a^b \int_c^d K(x_i,y_i,t,\varepsilon)\Phi_t(t,s)dtds\Big]^l_\omega\Big]^2, x_i,y_i \in [a,b]\times[c,d]$$

$$[\mathbb{E}(p)]^u_{i\omega} = \Big[\Big[\frac{d\Phi_t(x_i,p)}{dx}\Big]^u_\omega - \frac{1}{\varepsilon}\Big[\sum_{x_i,y_i,\in D} F(x_i,y_i,\Phi_t(x_i,p),\varepsilon) + \int_a^b \int_c^d K(x_i,y_i,t,\varepsilon)\Phi_t(t,s)dtds\Big]^u_\omega\Big]^2, x_i,y_i \in [a,b]\times[c,d]$$

where $\{x_i\}_{i=1}^g$ and $\{y_i\}_{i=1}^g$ are discrete points belong to $[a,b],[c,d]$ respectively $[E(p)]^l_{i\omega}$ and $[E(p)]^u_{i\omega}$ are the squared error measures associated with the respective lower and upper fuzzy bounds of the ω
$\frac{d[\Phi_t(x,p)]^l_\omega}{dx} = (x-a)\frac{d[\Phi_t(x,p,\varepsilon)]^l_\omega}{dx} + [OUT(x,p,\varepsilon)]^l_\omega$
$\frac{d[\Phi_t(x,p)]^u_\omega}{dx} = (x-a)\frac{d[\Phi_t(x,p,\varepsilon)]^u_\omega}{dx} + [OUT(x,p,\varepsilon)]^u_\omega$
Now it is easy to evaluate

$$[\mathbb{E}(p)]^l_{i\omega} = \Big[(x-a)\big(\frac{d[OUT(x,p,\varepsilon)]^l_\omega}{dx} + [OUT(x,p,\varepsilon)]^l_\omega\big) - \frac{1}{\varepsilon}\Big[\Big[\sum_{x_i,y_i,\in D} F(x_i,\Phi_t(x_i,p),\varepsilon) + \int_a^{b_i}\int_c^{d_i} K(x_i,y_i,t,s,\varepsilon)([A]^l_\omega + (t-a)[OUT(x,p,\varepsilon)\Big)]^l_\omega)dtds\Big]^l_\omega\Big]\Big]^2$$

$$[\mathbb{E}(p)]^u_{i\omega} = \Big[(x-a)\big(\frac{d[OUT(x,p,\varepsilon)]^u_\omega}{dx} + [OUT(x,p,\varepsilon)]^u_\omega\big) - \frac{1}{\varepsilon}\Big[\Big[\sum_{x_i,y_i,\in D} F(x_i,y_i,\Phi_t(x_i,p),\varepsilon) + \int_a^{b_i}\int_c^{d_i} K(x_i,y_i,t,s,\varepsilon)([A]^u_\omega + (t-a)[OUT(x,p,\varepsilon)\Big)]^u_\omega)dtds\Big]^u_\omega\Big]\Big]^2$$

The flowchart of neuro fuzzy system (NFS) which explains the implementation of the algorithm given in figure (2), where J is Jacobian matrix, J^T = transpose of a Jacobian matrix, μ is combination coefficient , I is the identity matrix and $e_{n,s}$ denotes the training error at output s when the input n is applied, and it is defined as
$e_{n,s} = d_{n,s} - t_{n,s}$
where d represents the desired output vector and t denotes the actual output vector.

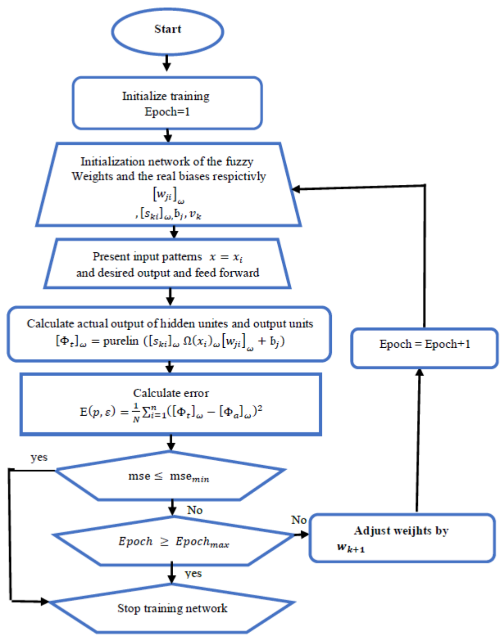

Figure 2: Flowchart of NFS

6. Numerical illustrations

We give examples to illustrate the technique of MATLAB 7.12 programs that display the characteristics and features of the suggested design. a PNFS to resolve FSPFIDEs, an PNFS with several layers, a single-input neural network with one hidden layer of seven neurons (neurons) were used. The suggested method dramatically improves accuracy and convergence speed thus cutting down on the quantity of iterations needed for each the goal, according to certain numerical experiments we show utilizing the NFS.

Example: In this problem, we consider the following two dimension FSPFIDE
$\varepsilon\Phi_x'(x,y) - \int_0^1\int_0^1 (t+s)\Phi(t,s)dtds = \frac{(y+1)}{\varepsilon} - \frac{\varepsilon x^3y^3}{2} - \frac{\varepsilon x^2y^3}{2}, x,y \in [a,b].[c,d], 0 < \varepsilon \ll 1, \omega \in [0,1]$,
with I.C: $[\Phi(0,y)]_\omega = [0,0]_\omega$
which has an explicit solution given by

$$\Phi(x,y) = [\omega - 1, 1 - \omega](xy + x)$$

The fuzzy trial solution

$$[\Phi_t(x,y)]_\omega = [[\Phi_t^l(x,y)]_\omega, [\Phi_t^u(x,y)]_\omega],$$

$[\Phi_t(x,p)]_\omega = [A]_\omega + (x-a)[O_{utp}(x,p,\varepsilon)]_\omega$ Then we get
$[\Phi_t(x,y)]_\omega^l = x[O_{utp}(x,p,\varepsilon)]_\omega^l$
$[\Phi_t(x,y)]_\omega^u = x[O_{utp}(x,p,\varepsilon)]_\omega^u$
Where $O_{utp}(x,p,\varepsilon)$ is the output of a feedforward PNFS with a single input unit corresponding to x and p is defined, and the associated error to be minimized is given by

$E(p) = \sum_{i=1}^{g}([E(p)]_{i\omega}^{l}, ([E(p)]_{i\omega}^{u}))$
where

$$[E(p)]_{i\omega}^{l} = \Big[\Big[\frac{d\Phi_t(x_i, y_i, p)}{dx}\Big]_{\omega}^{l} - \frac{1}{\varepsilon}\Big[\sum_{x_i, y_i, \in D} F(x_i, y_i, \Phi_t(x_i, p), \varepsilon) + \int_0^1 \int_0^1 K(x_i, y_i, t, s, \varepsilon)\Phi_t(t, s)dtds\Big]_{\omega}^{l}\Big]^2, x_i, y_i \in [a, b] \times [c, d]$$

$$[E(p)]_{i\omega}^{u} = \Big[\Big[\frac{d\Phi_t(x_i, y_i, p)}{dx}\Big]_{\omega}^{u} - \frac{1}{\varepsilon}\Big[\sum_{x_i, y_i, \in D} F(x_i, y_i, \Phi_t(x_i, p), \varepsilon) + \int_0^1 \int_0^1 K(x_i, y_i, t, s, \varepsilon)\Phi_t(t, s)dtds\Big]_{\omega}^{u}\Big]^2, x_i, y_i \in [a, b] \times [c, d]$$

$\frac{d[\Phi_t(x,p)]_{\omega}^{l}}{dx} = x\frac{d[O_{utp}(x,p,\varepsilon)]_{\omega}^{l}}{dx} + [O_{utp}(x, p, \varepsilon)]_{\omega}^{l}$
$\frac{d[\Phi_t(x,p)]_{\omega}^{u}}{dx} = x\frac{d[O_{utp}(x,p,\varepsilon)]_{\omega}^{u}}{dx} + [O_{utp}(x, p, \varepsilon)]_{\omega}^{u}$
Now it is easy to evaluate

$$[E(p)]_{i\omega}^{l} = \Big[x\frac{d[O_{utp}(x, p, \varepsilon)]_{\omega}^{l}}{dx} + [O_{utp}(x, p, \varepsilon)]_{\omega}^{l} - \frac{1}{\varepsilon}\Big[\sum_{x_i, y_i, \in D}\Big[\Big(\frac{(y+1)}{\varepsilon} - \frac{\varepsilon x^3 y^3}{2} - \frac{\varepsilon x^2 y^3}{2}\Big) + \int_0^1 \int_0^1 (t+s)\Big(t[O_{utp}(x, p, \varepsilon)\Big)]_{\omega}^{l}\Big]\Big]^2$$

$$[E(p)]_{i\omega}^{u} = \Big[x\frac{d[O_{utp}(x, p, \varepsilon)]_{\omega}^{u}}{dx} + [O_{utp}(x, p, \varepsilon)]_{\omega}^{u} - \frac{1}{\varepsilon}\Big[\sum_{x_i, y_i, \in D}\Big[\Big(\frac{(y+1)}{\varepsilon} - \frac{\varepsilon x^3 y^3}{2} - \frac{\varepsilon x^2 y^3}{2}\Big) + \int_0^1 \int_0^1 (t+s)\Big(t[O_{utp}(x, p, \varepsilon)\Big)]_{\omega}^{u}\Big]\Big]^2$$

Consequently, Evaluation of the function is simplet needs to be reduced for this state:
$\mathbb{E} = \sum_{i=1}^{11}(\mathbb{E}_{i\omega}^{l} + \mathbb{E}_{i\omega}^{u})$
The tables of errors demonstrate the precision and velocity of the suggested approach, while the table and figures that follow display the estimated answer and compare it with the precise solution. The tables of errors demonstrate the precision and velocity of the suggested approach.

input		"Analytic solution"		"Solution of PFNN $x_t(x)$"	
x	y	$[\Phi_a(x)]_{\omega}^{l}$	$[\Phi_a(x)]_{\omega}^{u}$	$[\Phi_t(x)]_{\omega}^{l}$	$[\Phi_t(x)]_{\omega}^{u}$
0	0	0.000000000000	0.000000000000	0.000000000000	0.000000000000
0.1	0.1	-0.055000000000	0.055000000000	-0.055000054389	0.055000043870
0.2	0.2	-0.120000000000	0.120000000000	-0.120000221876	0.120000005401
0.3	0.3	-0.195000000000	0.195000000000	-0.195000876541	0.195000006659
0.4	0.4	-0.280000000000	0.280000000000	-0.280070055430	0.280005611287
0.5	0.5	-0.375000000000	0.375000000000	-0.375000065433	0.375000033281
0.6	0.6	-0.480000000000	0.480000000000	-0.480004775342	0.480077432054
0.7	0.7	-0.595000000000	0.595000000000	-0.595000000328	0.595000398165
0.8	0.8	-0.720000000000	0.720000000000	-0.720004431876	0.720000565543
0.9	0.9	-0.855000000000	0.855000000000	-0.855000458856	0.855000555421
1	1	-1.000000000000	1.000000000000	-1.000000000000	1.000000000000

Table 1: The analytic and PNFS solution of the example where $\varepsilon = 10^{-6}, \omega = 0.5$

The error $[E(x)]_\omega = \lvert[\Phi_a(x)]_\omega - \Phi_t(x)]_\omega\rvert$	
$[error]^l_\omega$	$[error]^u_\omega$
0	0
5.43886999951337E-08	5.43886999951337E-08
2.21875999989463E-07	5.40089999112592E-09
8.76540900002221E-07	6.65909999364445E-09
0.0000700554299999934	0.0000056112869999958
6.54328999871723E-08	3.32810000114314E-08
4.77534230003407E-06	0.0000774320540000151
3.28000071547763E-10	3.98165000081718E-07
4.43187599996619E-06	5.65542999875213E-07
4.58856330043389E-07	5.55420999970302E-07
0	0
MSE= 4.95124134397887E-09	MSE= 6.02799775129661E-09

Table 2: Accuracy of the example's solutions (2.8) where $\varepsilon = 10^{-6}, \omega = 0.5$

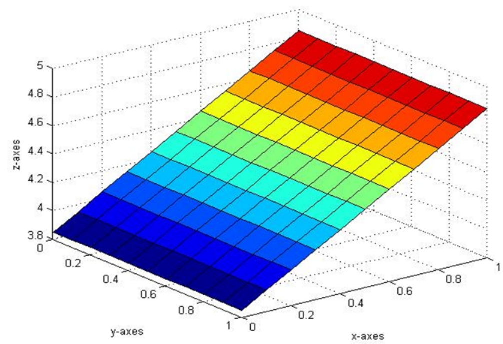

Figure 3: Comparison between analytic and neural-based solutions of the example with $\varepsilon = 10^{-6}$

7. Conclusions

In this paper, we used AI to solve a type of integral equation, namely FSPFIDEs, by designing an ANN consisting of one hidden layer and two input/output layers, with a hyperbolic activation function and the LM training algorithm. We proved the accuracy of the design by selecting a number of examples, and by comparing them, we demonstrated the accuracy and speed of the solution.

References

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

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. Salman, G.S., Al-Abrahemee, K.M.M., Designing feed forward fully fuzzy neural network to solve fuzzy singular perturbed Volterra integro-differential equations , Journal of Interdisciplinary Mathematics, 2025, 28(1), pp. 139–146. https://doi.org/10.47974/JIM-1804.

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., 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. Al-Abrahemee, K.M.M., Yousif, M.S. , A new parallel numerical algorithm for solving singular Perturbation problems in partial differential equations, Journal of Prime Research in Mathematics, 17(2), (2021), 111-122.ISSN: 1817-3462E (Online) 1818-5495 (Print).

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. Khalid Mindeel Mohammed Al-Abrahemee, Alaa Kamil Jabber, A parallel numerical training algorithm for solving of a class of delay partial differential equations, AIP Conf. Proc. 2845, 050041 (2023). ttps://doi.org/10.1063/5.0157086.

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.

Thaer Ubaid Al-Jubouri,
Department of Mathematics,
College of Education, University AL-Qadisiyha,
Iraq.
E-mail address: `Email:thaerkreade7@gmail.com`

and

Khalid Mindeel Mohammed Al-Abrahemee,
Department of Mathematics,
College of Education, University AL-Qadisiyha,
Iraq.
E-mail address: `khalid.mohammed@qu.edu.iq`