DOI: 10.24996/ijs.2024.65.11.31

ISSN: 0067-2904

Solving Singular Perturbation Problems With Initial and Boundary Conditions By Using Modified Neuro System

Tabark Aqeel Al-Janabi*, Khalid Mindeel Mohammed Al-Abrahemee
Department of Mathematics, College of Education, University of Al-Qadisiyah, Al-Qadisiyah, Iraq

Received: 11/5/2023 Accepted: 8/10/2023 Published: 30/11/2024

Abstract
The aim of this paper is to design a neural network for solving the singular perturbation problems by using neural networks. The modified neuro system using a polynomial of second degree is to replace each component in the training set. The foundation of this approach is to swap off each x in the input vector training set $\vec{x_j} = (x_1, x_2, \dots, x_n)$, $x_j \in [a, b]$, the polynomial will be as $\xi(x) = \frac{\lambda}{2}(x^2 + x + 1)$, $\lambda \in (a, b)$. The appropriate value is determined within a certain range, which has a significant impact on the accuracy of the solution. The numerical results show that the modified neuro system method is better and more accurate than usual artificial neural network method, the main reason for this point is connected with the chosen value of . Finally, a method of updating the neural network is clarified by the numerical results of some examples that are compared to the usual artificial neural network method and through which the accuracy of the solution and the rapidity of convergence is proved.

Keywords: singular perturbation problems, neural networks, training set

حل مشاكل الاضطراب الفردي مع الشروط الأولية والحدودية باستخدام نظام عصبي معدل

تبارك عقيل الجنابي*, خالد منديل محمد الابراهيمي
قسم الرياضيات, كلية التربية, جامعة القادسية, القادسية, العراق

الخلاصة
الهدف من هذا البحث هو تصميم شبكة عصبية لحل مشاكل الاضطراب المفرد باستخدام الشبكات العصبية. النظام العصبي المعدل باستخدام متعددة الحدود من الدرجة الثانية لاستبدال كل مكون في مجموعة التدريب. أساس هذا النهج هو تبديل كل x في مجموعة التدريب بمتجه الادخال $x_j \in$ ، $=(x_1, x_2, \dots, x_n)$ $\vec{x_j}$ $[a, b]$ ، متعددة الحدود التي ستكون بالشكل $\xi(x) = \frac{\lambda}{2}(x^2 + x + 1)$ $\lambda \in (a, b)$ ، . تم تحديد القيمة المناسبة ضمن نطاق معين، مما له تأثير كبير على دقة الحل. أظهرت النتائج العددية أن طريقة النظام العصبي المعدل أفضل وأكثر دقة من طريقة الشبكة العصبية الاصطناعية المعتادة، وبطبيعة الحال، السبب الرئيسي لهذه النقطة يرتبط بالقيمة المختارة لـ λ .أخيراً تم توضيح طريقة تحديث الشبكة العصبية من خلال النتائج العددية لبعض الأمثلة والتي تم مقارنتها بطريقة الشبكة العصبية الاصطناعية المعتادة والتي من خلالها تم إثبات دقة الحل وسرعة التقارب.

*Email: kkak1962@yahoo.com

1. Introduction

Nowadays, a new branch of computational science has emerged which integrates several techniques to solve many problems that are not easily stated without an algorithmic typical focus. In one form or another, these approaches are inspired by the imitation of biological systems' behavior that are done in a fashion which is either more or less intelligent. It is a brand-new approach to computing known as artificial intelligence which uses a variety of techniques to manage the uncertainty and imprecision that arise when attempting to solve problems that relate to the actual world while these techniques provide effective solutions that are simple to apply. The one of these methods are Artificial neural networks (ANNs) [1]. Differential equations are used to formulate many issues, and the nonlinear terms only depend on certain dependent variable derivatives and a tiny value parameter ε. The typical view of these weakly nonlinear issues is that they are perturbations of the corresponding linear differential equations [2]. Applications of the perturbed issues for differential equations are fairly common, and they have received a lot of attention recently. Singular perturbation issues frequently arise in a variety of fields of applied mathematics, such as fluid dynamics, elasticity, chemical reactor theory, aerodynamics, magneto hydrodynamics, and plasma dynamics [3]. Recently, large range of books and papers that are outlining numerous approaches to solving SPPs have been published. Among these, Lagerstrom and Casten [4]. A class of singular perturbation problems with certain applications in fluid dynamics is solved using the perturbation technique. Amiraliyev [5] gave the numerical solution of the initial condition of the second order linear singly perturbed problem. Arianov et al. [6] studied of a perturbation technique application with a few perturbation parameters. There are many studies on solving perturbation problems related to artificial intelligence and open learning such as artificial neural networks. Artificial neural networks (ANNs) is a calculation method that builds several processing units based on interconnected connections. The network consists of an arbitrary number of cells or nodes or units or neurons that connect the input set to the output. It is a part of a computer system that mimics how the human brain analyzes and processes data. Self-driving vehicles, character recognition, image compression, stock market prediction, risk analysis systems, drone control, welding quality analysis, computer quality analysis, emergency room testing, oil and gas exploration and a variety of other applications all use artificial neural networks. Predicting consumer behavior, creating and understanding more sophisticated buyer segments, marketing automation, content creation and sales forecasting are some applications of the ANN systems in the marketing [7]. In fact, ANNs 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. For the user knowledge, it is necessary to implement NNs successfully that are substantially lower than others [8] [9]. Artificial neural networks have been used to solve problems in various educational and industrial fields [10] [11] [12] [13]. Dash and Daripa [14] have been released and presented analyses of a singularly perturbed Boussinesq equation using analytical and numerical methods. Hunter [15] used the numerical method to address a particular class of PPs that demonstrates the inadequacy of traditional discretization methods. Shikongo [3] created and put into practice some unique numerical techniques for some non-linear SPPs. Valanarasu and Ramanujam [16] proposed a numerical approach to solve ordinary differential equations (ODEs) second-order SPP with two points boundary conditions (BCs) ,as well as there are many papers on the use of modifying the neural network to solve differential equations by modifying the training algorithm or some parameters associated with the network design. Also, it has been used (MANN) for solving SPPs. This approach is according to substituting every x on the input vector training set with the first-degree polynomial [9]. In this paper, the study is different from the modernization methods that are previously used. The aim of this study is to present a modified method for finding the numerical solutions of SPPs for ODEs by using a modified neuro system (MNS_1) which will be explained in the next sections.

2. Perturbation problems

The perturbed differential equation problems (PPs) are a common occurrence in applications that have received a great deal of attention recently. As a result, PPs are categorized into two categories based on their location: singular perturbed problems (SPPs) and regular perturbed problems. These issues are known to depend on a small positive parameter ε in a way that causes the solution to have a multiscale nature that means there are thin transition layers where the answer changes quickly [5].

Differential equations with the highest derivative is multiplied by a small parameter ε are known as singly perturbed differential equations. SPPs for ODEs in their general form, which have a small positive parameter ε, $0 < \varepsilon << 1$, have the following form (in case of the second order):

$$\psi''(\mathrm{x}) = F(\mathrm{x}, \psi, \psi', \varepsilon), \mathrm{x} \in [a, b]. \quad (1)$$

Where F is a generalized nonlinear function of their arguments, and
$F(\mathrm{x}, \psi, \psi', \varepsilon) \epsilon C^3([a, b] \times R^2 \times (0,1))$,
$\frac{\partial F}{\partial \varepsilon}(\mathrm{x}, \psi, \psi', \varepsilon) \neq 0, (\mathrm{x}, \psi, \psi', \varepsilon) \epsilon ([a, b] \times R^2 \times (0,1))$

Remark: Suppose that there is only one small, positive parameter in our problem. ε $(0 < \varepsilon \ll 1)$, P_ε represents the problem. What occurs if $\varepsilon \to 0$? , the reduced problem is had by P_0. Under reasonable assumptions, the connection will be investigated between the P_ε and P_0 solutions. A perturbation problem (1) is called SPP, if $\varepsilon \to 0$, the solution $\psi_\varepsilon(\mathrm{x})$ converges to $\psi_0(\mathrm{x})$ only at some x-interval, but it does not for the full time period, thus giving rise to the "boundary layers" phenomena at both endpoints [17].

3. Mean squared error (MSE): It measures the amount of error in statistical models. It assesses the average squared difference between the observed and predicted values. When a model has no error, the MSE equals zero. As model error increases, its value increases. The mean squared error is also known as the mean squared deviation (MSD).

The formula for MSE is the following: $MSE = \frac{\sum (y_i - \hat{y}_i)^2}{n}$

Where : y_i is the i^{th} observed value , $\hat{y}$ is the corresponding predicted value and n is the number of observations.

4. Architectural structure

In this section, we will employ the neural networks based on the polynomial. $\xi(\mathrm{x}) = \frac{\lambda}{2}(\mathrm{x}^2 + \mathrm{x} + 1), \lambda \in (0,1)$ to solve the singular perturbation problems. The neural network is a three-layer feed forward (NN) where the connections weights, biases, and targets are given as real numbers and the inputs are also given as real numbers. The basic structural architecture of this technique (MNS_1) includes input layers one is a hidden layer and an output layer. Here, the dimension is indicated by the amount of neurons in each layer, which is $n \times m \times s$, where n denotes the number of neurons in the input layer, m is the number of neurons in the hidden layer and s is the number of neurons in the output layer. The architecture of the model shows the transformation of the n inputs $(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{i}, \mathrm{x}_{\mathrm{i}+1}, \ldots, \mathrm{x}_\mathrm{n})$ into the s outputs $(\psi_1, \psi_2, \ldots, \psi_k, \psi_{k+1}, \ldots, \psi_s)$ throughout the m hidden neurons

$(\mathrm{Hid}_1, \mathrm{Hid}_2, \dots, \mathrm{Hid}_j, \mathrm{Hid}_{j+1}, \dots, \mathrm{Hid}_m)$ where the cycles represent the neurons in each layer. Let b_j , ϑ_k , w_{ji} and , s_{kj} be the bias for the neurons Hid_j , the bias for the neurons ψ_k , the weights connecting the neurons x_i to the neurons Hid_j and the weights connecting the neurons Hid_j to the neurons ψ_k, respectively. When the n-dimensional input vector $(x_1, x_2, \dots, x_i, x_{i+1}, \dots, x_n)$ is presented to the neural network. Its input and output relations can be written as the following algorithm of the modified neuro system (MNS_1):
Where x and ψ are the input and output, respectively.
Step 1: Start
Step 2: x_i represent the input units

$$x_i = \xi(x_i) = \frac{\lambda}{2}(x_i^2 + x_i + 1),\ i = 1, \dots, n,\ \lambda \in (0,1)$$

Step 3: Hidden units

$$\mathrm{Hid}_j = T(\mathrm{Netw}_j),\ j = 1,2, \dots, m$$

$$\mathrm{Netw}_j = \sum_{i=1}^{n} x_i w_{ji} + b_j$$

$$= \sum_{i=1}^{n} \frac{\lambda}{2}\left(x_i^2 + x_i + 1\right) w_{ji} + b_j$$

where w_{ji} are the input layer's weight parameter, which jth is the unit in the hidden layer, b_j is an jth bias for the hidden layer unit.
Step 4: Output units

$$\mathrm{Out}_k(\xi(x),p,\varepsilon) = T\left(\mathrm{Netw}_k\right),\ k=1,2,\dots,s$$

$\mathrm{Netw}_k = \sum_{j=1}^{m} s_{kj}\mathrm{Hid}_j + \vartheta_k$, where T is the hyperbolic tangent activation function, $\mathrm{Out}(\xi(x),p,\varepsilon)$ of the output network and s_{kj} is a weight parameter from jth unit in the hidden layer to output layer.
Step 5: Calculation of the trial solution ψ_k.
Step 6: Stop.

Theorem: Let a and b be positive real numbers, If $x \in [a, b]$, then the appropriate value of λ can be determined to guarantee that $\xi(x) = \frac{\lambda}{2}(x + 1), \xi(x) \in (a, b)$ such that: $\frac{2a}{a+1} < \lambda < \frac{2b}{b+1}$.

Proof: Since $x \in [a, b]$, and since $\xi(x) = \frac{\lambda}{2}(x + 1)$.
Then $\xi(x) = \frac{\lambda}{2}[a + 1, b + 1] = [\frac{\lambda}{2}(a + 1), \frac{\lambda}{2}(b + 1)]$ is obtained if $\frac{\lambda}{2}(a + 1) = a$ is considered: $\lambda = \frac{2a}{a+1}$ and if we consider $\frac{\lambda}{2}(b + 1) = b$, then $\lambda = \frac{2b}{b+1}$ can be get.
Therefore, if we consider $\lambda > \frac{2a}{a+1}$ and $\lambda < \frac{2b}{b+1}$,

then $\xi(x) = [\frac{\lambda}{2}(a + 1), \frac{\lambda}{2}(b + 1)] \in \left(\frac{a}{a+1}(a + 1), \frac{b}{b+1}(b + 1)\right) = (a, b)$.

Therefore, we have $\xi(x) \in (a, b)$ if $\frac{2a}{a+1} < \lambda < \frac{2b}{b+1}$.

5. Illustration of MNS_1 for solving SPP

5.1 Solution of the second-order SPPs with IC

For the second-order of SPPs that is considered by :

$$\varepsilon\psi'' = F(x, \psi, \psi', \varepsilon),\ x \in [a, b],\quad 0 < \varepsilon << 1, \tag{2}$$

$\psi(a) = A$, $\psi'(a) = B$.

where ψ is a function with derivative ψ', A and B are real numbers.
The trial function will be in the form:

$$\psi_t(x,p,\varepsilon) = A + B(x-a) + (x-a)^2 Out(\xi(x),p,\varepsilon) . \quad (3)$$

The conditions in eq. (2) are intentionally satisfied by this solution,
and $\{x_i\}_{i=1}^{g}$ are discrete points that fall within the interval $[a,b]$.
Now, we differentiate the trial function $\psi_t(x,p,\varepsilon)$ in eq.(3) to find the amount of error, then we get the following:

$$\frac{\partial\psi_t(x,p,\varepsilon)}{\partial x} = B + 2(x-a)Out(\xi(x),p,\varepsilon) + (x-a)^2\frac{\partial Out(\xi(x),p,\varepsilon)}{\partial x} , \quad (4)$$

$$\frac{\partial^2\psi_t(x,p,\varepsilon)}{\partial x^2} = 2[Out(\xi(x),p,\varepsilon)] + 4(x-a)\frac{\partial Out(\xi(x),p,\varepsilon)}{\partial x} + (x-a)^2\frac{\partial^2 Out(\xi(x),p,\varepsilon)}{\partial x^2} , \quad (5)$$

Where $Out(\xi(x),p,\varepsilon) = \sum_{j=1}^{m} s_j\, T(\xi(x)\, w_j + b_j)$, (6)

$$\frac{\partial Out(\xi(x),p,\varepsilon)}{\partial x} = \sum_{j=1}^{m}\frac{\lambda}{2}(2x_i + 1)w_j s_j\, T'(\xi(x)\, w_j + b_j) , \quad (7)$$

$$\frac{\partial^2 Out(\xi(x),p,\varepsilon)}{\partial x^2} = \sum_{j=1}^{m}\frac{\lambda^2}{2}\, w_j^2\,(2x+1)^2\, s_j\, T''(\xi(x)\, w_j + b_j) + \lambda\, w_j\, s_j\, T'(\xi(x)\, w_j + b_j). \quad (8)$$

5.2 Solution for system of SPPs

Consider the system of K first-order ODEs :

$$\varepsilon\psi_i{}' = F_i(x, \psi_1, \psi_2, \dots, \psi_K, \varepsilon) , 0 < \varepsilon << 1 \quad (9)$$

with $\psi_i(0) = A_i$, $i = 1, 2, \dots, K$. We consider one ANN for each trial solution ψ_{t_i}, $i = 1, 2, \dots, K$ which can be written as follows:

$$\psi_{t_i}(x,p,\varepsilon) = A_i + x\, Out_i(\xi(x),p_i,\varepsilon) . \quad (10)$$

Additionally, to reduce the amount of error:

$$\min_{\vec{\mathcal{P}}} \sum_{\vec{\varkappa}_i \in \hat{D}} \mathcal{F}\left(\left(\vec{\varkappa}_i, \varepsilon, \psi_t(\vec{\varkappa}_i, \vec{\mathcal{P}}, \varepsilon), \psi'_t(\vec{\varkappa}_i, \vec{\mathcal{P}}, \varepsilon), \psi''_t(\vec{\varkappa}_i, \vec{\mathcal{P}}, \varepsilon), \dots\right)\right)^2 \quad (11)$$

5.3 Solution of the second-order SPPs with B.C

Consider the second-order of SPPs for ODEs

$$\varepsilon\psi'' = F(x, \psi(x), \psi', \varepsilon) \quad ,x \in [a,b]. \quad (12)$$

Where ε is the perturbation $(0 < \varepsilon << 1)$ with the boundary conditions : $\psi(a) = A, \psi(b) = B$. For this problem, the trial solution is as follows :

$$\psi_t(x,p,\varepsilon) = \frac{bA-aB}{b-a} + \frac{B-A}{b-a}x + (x-a)(x-b)[Out(\xi(x),p,\varepsilon)]. \quad (13)$$

Now we differentiate the trial function $\psi_t(x,p,\varepsilon)$ in eq.(13), then we obtain:

$$\frac{d\psi_t(x_i,p,\varepsilon)}{dx} = \frac{B-A}{b-a} + (x-a)(x-b)\frac{dOut(\xi(x),p,\varepsilon)}{dx} + \big(2x-(a+b)\big)[Out(\xi(x),p,\varepsilon)] . \quad (14)$$

$$\frac{d^2\psi_t(x_I,p,\varepsilon)}{d\varkappa^2} = (x-a)(x-b)\frac{d^2Out(\xi(x),p,\varepsilon)}{dx^2} + 2\big(2x-(a+b)\big)\frac{dOut(\xi(x),p,\varepsilon)}{dx} + 2Out(\xi(x),p,\varepsilon). \quad (15)$$

Where $Out(\xi(x),p,\varepsilon)$ is the output of the feed forword MNS_1 with one input for x and parameter p. Hence,

$$Out(\xi(x),p,\varepsilon) = \sum_{j=1}^{m} s_j\, T(\xi(x)\, w_j + b_j) , \quad (16)$$

$$\frac{\partial Out(\xi(x),p,\varepsilon)}{\partial x} = \sum_{j=1}^{m}\frac{\lambda}{2}(2x_i + 1)w_j s_j\, T'(\xi(x)\, w_j + b_j), \quad (17)$$

$$\frac{\partial^2 Out(\xi(x),p,\varepsilon)}{\partial x^2} = \sum_{j=1}^{m}\frac{\lambda^2}{2}\, w_j^2\,(2x+1)^2\, s_j\, T''(\xi(x)\, w_j + b_j) + \lambda\, w_j\, s_j\, T'(\xi(x)\, w_j + b_j). \quad (18)$$

The amount of the error which must be minimized is given as follows:

$$\mathbb{E}_i(p,\varepsilon) = \sum_{i=1}^{g}\left[\frac{d^2\psi_t(x_i,p,\varepsilon)}{d\varkappa^2} - \frac{1}{\varepsilon}\left[F\left[x_i, \psi_t(x_i,p,\varepsilon), \frac{d\psi_t(x_i,p,\varepsilon)}{dx}, \varepsilon\right)\right]\right]^2 . \quad (19)$$

Where $\{x_i\}_{i=1}^{g} \in [a,b]$ are discrete points ,respectively. Then, eq.(19) can be rewritten as:

$$\mathbb{E}_i(p,\varepsilon) = \sum_{i=1}^{g}\Bigg[(x_i-a)(x_i-b)\frac{d^2Out(\xi(x),p,\varepsilon)}{dx^2} + 2\big(2x_i-(a+b)\big)\frac{dOut(\xi(x),p,\varepsilon)}{dx} + 2Out(\xi(x),p,\varepsilon) - \frac{1}{\varepsilon}F\Bigg(x_i, \frac{bA-aB}{b-a} + \frac{B-A}{b-a}x_i + (x_i-a)(x_i-b)Out(\xi(x),p,\varepsilon), \frac{B-A}{b-a} + (x_i-a)(x_i-b)\frac{d(Out(\xi(x),p,\varepsilon))}{dx} + \big(2x_i-(a+b)\big)Out(\xi(x),p,\varepsilon)\Bigg)\Bigg]^2 . \quad (20)$$

6. Numerical illustrations

In this section, some numerical results and the resolution of several models SPPs in every instance have been used to suggest the employing multiple-layer perceptron, which consists of one input of 7 hidden units in one hidden layer and one linear output unit. As the analytical solution is already known $\psi_a(x)$ to each test problem, so we can determine the accuracy of the solutions and that is found by computing the deviation : $E(x,p,\varepsilon) = |\psi_t(x,p,\varepsilon) - \psi_a(x,p,\varepsilon)|$.

Example1: Consider the following linear system of SPPs:

$\frac{d\psi_1}{dx} = -2\psi_1(x) + \psi_2(x) + 2sinx$,

$\varepsilon\frac{d\psi_2}{dx} = -(1+2\varepsilon)\psi_1(x) + (1+\varepsilon)(\psi_2(x) - cosx + sinx)$,

$\psi_1(0) = 2$,$\psi_2(0) = 3$.

The exact solution of this problem is given by the following:

$\psi_1(x) = 2e^{-x} + sinx$, $\psi_2(x) = 2e^{-x} + cosx$.

Then, the trial solutions are

$\psi_{1t}(x,p,\varepsilon) = 2 + x\mathrm{Out}(\xi(x),p,\varepsilon)$, $\psi_{2t}(x,p,\varepsilon) = 3 + x\mathrm{Out}(\xi(x),p,\varepsilon)$.

The MNS_1 is trained using a grid of ten equidistant points in the interval $[0,1]$ that means the input vector $\vec{x}$ (training set) is: $\vec{x} = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. Now, to find the error function E that must be minimized for this problem, the following steps have to be applied:

$\frac{\partial\psi_{1t}(x,p,\varepsilon)}{\partial x} = \mathrm{Out}(\xi(x),p,\varepsilon) + x\frac{\partial \mathrm{Out}(\xi(x),p,\varepsilon)}{\partial x}$,

$\frac{\partial\psi_{2t}(x,p,\varepsilon)}{\partial x} = \mathrm{Out}(\xi(x),p,\varepsilon) + x\frac{\partial \mathrm{Out}(\xi(x),p,\varepsilon)}{\partial x}$.

Then, we get: $\mathbb{E}_i(p,\varepsilon) = \sum_{i=1}^{11}\left[\frac{\partial\psi_{1t}(x_i,p,\varepsilon)}{\partial x} - (-2\psi_{1t}(x_i) + \psi_{2t}(x_i) + 2sinx_i)\right]^2 + \left[\frac{\partial\psi_{2t}(x_i,p,\varepsilon)}{\partial x} - \frac{1}{\varepsilon}\left(-(1+2\varepsilon)\psi_{1t}(x_i) + (1+\varepsilon)(\psi_{2t}(x_i) - cosx_i + sinx_i)\right)\right]^2$,

$\mathbb{E}_i(p,\varepsilon) = \sum_{i=1}^{11}\Big[\mathrm{Out}(\xi(x_i),p,\varepsilon) + x_i\frac{\partial \mathrm{Out}(\xi(x_i),p,\varepsilon)}{\partial x} - \big(-2(2 + x_i\,\mathrm{Out}(\xi(x_i),p,\varepsilon)) + (3 + x_i\,\mathrm{Out}(\xi(x_i),p,\varepsilon) + 2sinx_i)\big)\Big]^2 + \Big[\mathrm{Out}(\xi(x_i),p,\varepsilon) + x_i\frac{\partial \mathrm{Out}(\xi(x_i),p,\varepsilon)}{\partial x} - \frac{1}{\varepsilon}\big(-(1+2\varepsilon)2 + x_i\,\mathrm{Out}(\xi(x_i),p,\varepsilon) + (1+\varepsilon)(3 + x_i\mathrm{Out}(\xi(x_i),p,\varepsilon) - cosx_i + sinx_i)\big)\Big]^2$.

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

$\frac{\partial \mathrm{Out}(\xi(x),p,\varepsilon)}{\partial x} = \sum_{j=1}^{7}\frac{\lambda}{2}(2x_i + 1)w_j s_j\,T'(\xi(x)w_j + b_j)$.

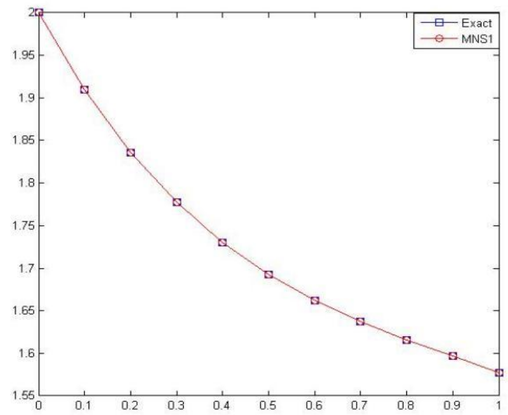
Therefore, we get: $\mathbb{E}_i(p,\varepsilon) = \sum_{i=1}^{11}\Big[\sum_{j=1}^{7} v_j\,T(\xi(x_i)w_j + b_j) + x_i\sum_{j=1}^{7}\frac{\lambda}{2}(2x_i + 1)w_j s_j\,T'(\xi(x_i)w_j + b_j) - \Big(-2\big(2 + x_i\sum_{j=1}^{7}s_j\,T(\xi(x_i)w_j + b_j)\big) + \big(3 + x_i\big(\sum_{j=1}^{7}s_j\,T(\xi(x_i)w_j + b_j) + 2sinx_i\big)\big)\Big)\Big]^2 + \Big[\big(\sum_{j=1}^{7}s_j\,T(\xi(x_i)w_j + b_j)\big) + x_i\sum_{j=1}^{7}\frac{\lambda}{2}(2x_i + 1)w_j s_j\,T'(\xi(x_i)w_j + b_j) - \frac{1}{\varepsilon}\Big(-(1+2\varepsilon)\big(2 + x_i\sum_{j=1}^{7}s_j\,T(\xi(x_i)w_j + b_j)\big) + (1+\varepsilon)\big(3 + x_i\big(\sum_{j=1}^{7}s_j\,T(\xi(x_i)w_j + b_j)\big) - cosx_i + sinx_i\big)\Big)\Big]^{2^2}$.

Since x in this example is between 0 and 1 and according to theorem, it requires to select $0 < \lambda < 1$, then it is selected $\lambda = 0.3$ and let $\varepsilon = 10^{-7}$.

The training set $\vec{x} = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$,

$\xi(x)$: 0.15 0.19 0.24 0.28 0.33 0.37 0.42 0.46 0.51 0.5 0.6 .

In Figures 1-2, the analytical and neural solutions found in the training set are shown by the feed forward MNS_1 trained using a grid of evenly spaced points in $[0,1]$. Then the oral results for MNS_1 , UANN, and train accuracy errors are shown in table 1.



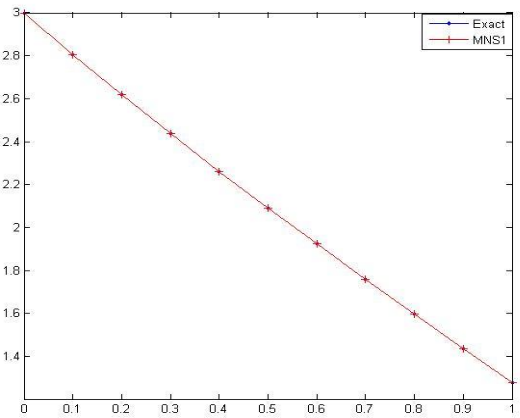


Figure 1: Analytic and MNS_1 of ψ_{1t} in example 1, with $\varepsilon = 10^{-7}$.

Figure 2: Analytic and MNS_1 of ψ_{2t} example 1, with $\varepsilon = 10^{-7}$**.**

Table 1: Analytic , MNS_1 solution and accuracy of the train of example 1 , $\varepsilon = 10^{-7}, \lambda = 0.3$.

Input x	**Analytic solution $\psi_{1a}(x)$**	**Analytic solution $\psi_{2a}(x)$**	**Solution of MNS_1 $\psi_{1t}(x)$ for training algorithm**	**Solution of MNS_1 $\psi_{2t}(x)$ for training algorithm**	**Accuracy of solutions of MNS_1 $E(x) = \lvert\psi_{1t}(x) - \psi_{1a}(x)\rvert$**	**Accuracy of solutions of MNS_1 $E(x) = \lvert\psi_{2t}(x) - \psi_{2a}(x)\rvert$**
0	2.000000000000000	3.000000000000000	2.000000000000000	3.000000000000000	0	0
0.1	1.909508252718750	2.804679001349940	1.909508242568540	2.804679066475460	1.01502E-08	6.51255E-08
0.2	1.836130836951020	2.617528083997210	1.836130855367320	2.617528065433560	1.84163E-08	1.85636E-08
0.3	1.777156648024780	2.436972930489040	1.777156646432750	2.436972938865240	1.59203E-09	8.3762E-09
0.4	1.730058434379930	2.261701086074160	1.730058431125850	2.261701086773570	3.25408E-09	6.99406E-10
0.5	1.692486858029470	2.090643881315640	1.692486834563250	2.090648853557800	2.34662E-08	4.97224E-06
0.6	1.662265745583090	1.922958887097730	1.662265763278740	1.922958899535780	1.76957E-08	1.2438E-08
0.7	1.637388294820510	1.758012794867310	1.637388294573130	1.758012954567930	2.4738E-10	1.59701E-07
0.8	1.616014019133970	1.595364637581610	1.616014552773150	1.595364688567980	5.33639E-07	5.09864E-08
0.9	1.596466229108680	1.434749287751860	1.596466266315740	1.434749288858950	3.72071E-08	1.10709E-09
1	1.577229867150780	1.276061188211020	1.577229865531570	1.276061185544570	1.61921E-09	2.66645E-09
The accuracy of the train		**Time**	**Epoch**	**Time**	**Epoch**	
		0:00:01	11	0:00:3	44	**MSE**=2.61343E-14 / **MSE**=2.25056E-12

Example 2: Consider the following second-order nonlinear SPP: $\varepsilon\psi'' + \psi' + \psi^2 = 0$ with the Dirichiet BC's : $\psi(0) = 0$, $\psi(1) = 1/2$.
The exact solution of this problem is given by : $\psi_t(\mathrm{x}) = \frac{1}{1+\mathrm{x}} - \frac{e^{-\mathrm{x}/\varepsilon}}{(1+\mathrm{x})^2}$.
Then, trial solutions are : $\psi_t(\mathrm{x}, p) = \frac{1}{2}\mathrm{x} + \mathrm{x}(\mathrm{x}-1)[\mathrm{Out}(\xi(\mathrm{x}), \mathrm{p}, \varepsilon)]$
The MNS_1 trained using a grid of ten equidistant points in the interval$[0.1]$ that means the input vector $\vec{\mathrm{x}}$ (training set) is:
$\vec{\mathrm{x}} = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$.
Now, to find the error function E that must be minimized for this problem, the following steps have to be applied:

$$\frac{\partial\psi_t(\mathrm{x,p},\varepsilon)}{\partial\mathrm{x}} = \frac{1}{2} + (2\mathrm{x}-1)[\mathrm{Out}(\xi(\mathrm{x}),\mathrm{p},\varepsilon)] + (\mathrm{x}^2-\mathrm{x})\frac{\partial\mathrm{Out}(\xi(\mathrm{x}),\mathrm{p},\varepsilon)}{\partial\mathrm{x}},$$

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

Then, we get the following: $\mathbb{E}_i(p,\varepsilon) = \sum_{i=1}^{11}\left[\frac{d^2\psi_t(\mathrm{x}_I,p,\varepsilon)}{d\varkappa^2} - \frac{1}{\varepsilon}[-\psi' - \psi^2]\right]^2$,

$$\mathbb{E}_i(p,\varepsilon) = \sum_{i=1}^{11}\Bigg[2[\mathrm{Out}(\xi(\mathrm{x}_i),\mathrm{p},\varepsilon)] + 2\left((2\mathrm{x}_i-1)\frac{\partial\mathrm{Out}(\xi(\mathrm{x}_i),\mathrm{p},\varepsilon)}{\partial\mathrm{x}}\right) + (\mathrm{x}_i{}^2-\mathrm{x}_i)\frac{\partial^2\mathrm{Out}(\xi(\mathrm{x}_i),\mathrm{p},\varepsilon)}{\partial\mathrm{x}^2} - \frac{1}{\varepsilon}\Big[-\Big(\frac{1}{2} + (2\mathrm{x}_i-1)[\mathrm{Out}(\xi(\mathrm{x}_i),\mathrm{p},\varepsilon)] + (\mathrm{x}_i{}^2-\mathrm{x}_i)\frac{\partial\mathrm{Out}(\xi(\mathrm{x}_i),\mathrm{p},\varepsilon)}{\partial\mathrm{x}}\Big) - \Big(\frac{1}{2}\mathrm{x}_i + \mathrm{x}_i(\mathrm{x}_i - 1)[\mathrm{Out}(\xi(\mathrm{x}_i),\mathrm{p},\varepsilon)]\Big)^2\Big]\Bigg]^2.$$

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

$$\frac{\partial\mathrm{Out}(\xi(\mathrm{x}),\mathrm{p},\varepsilon)}{\partial\mathrm{x}} = \sum_{\mathrm{j}=1}^{7}\frac{\lambda}{2}(2\mathrm{x_i}+1)\mathrm{w_j}\,\mathrm{s_j}\,\mathrm{T}'(\xi(\mathrm{x})\,\mathrm{w_j} + \mathrm{b_j}) \text{ , and}$$

$$\frac{\partial^2\mathrm{Out}(\xi(\mathrm{x}_i),\mathrm{p},\varepsilon)}{\partial\mathrm{x}^2} = \sum_{\mathrm{j}=1}^{7}\frac{\lambda^2}{2}\mathrm{w_j^2}(2\mathrm{x}+1)^2\mathrm{s_j}\,\mathrm{T}''\big(\xi(\mathrm{x})\,\mathrm{w_j} + \mathrm{b_j}\big) + \lambda\,\mathrm{w_j}\,\mathrm{s_j}\,\mathrm{T}'(\xi(\mathrm{x})\,\mathrm{w_j} + \mathrm{b_j}).$$

Therefore, we get:

$$\mathbb{E}_i(p,\varepsilon) = \sum_{i=1}^{11}\Bigg[\sum_{\mathrm{j}=1}^{7}\mathrm{s_j}\,\mathrm{T}(\xi(\mathrm{x_i})\,\mathrm{w_j}+\mathrm{b_j}) + 2\Big((2\mathrm{x}_i-1)\sum_{\mathrm{j}=1}^{7}\frac{\lambda}{2}(2\mathrm{x_i}+1)\mathrm{w_j}\,\mathrm{s_j}\,\mathrm{T}'(\xi(\mathrm{x_i})\,\mathrm{w_j}+\mathrm{b_j})\Big) + (\mathrm{x}_i{}^2-\mathrm{x}_i)\sum_{\mathrm{j}=1}^{7}\frac{\lambda^2}{2}\mathrm{w_j^2}(2\mathrm{x}+1)^2\mathrm{s_j}\,\mathrm{T}''\big(\xi(\mathrm{x_i})\,\mathrm{w_j}+\mathrm{b_j}\big) + \lambda\,\mathrm{w_j}\,\mathrm{s_j}\,\mathrm{T}'(\xi(\mathrm{x_i})\,\mathrm{w_j}+\mathrm{b_j}) - \frac{1}{\varepsilon}\Big[-\Big(\frac{1}{2} + (2\mathrm{x}_i-1)\big[\sum_{\mathrm{j}=1}^{7}\mathrm{s_j}\,\mathrm{T}(\xi(\mathrm{x_i})\,\mathrm{w_j}+\mathrm{b_j})\big] + (\mathrm{x}_i{}^2-\mathrm{x}_i)\sum_{\mathrm{j}=1}^{7}\frac{\lambda}{2}(2\mathrm{x_i}+1)\mathrm{w_j}\,\mathrm{s_j}\,\mathrm{T}'(\xi(\mathrm{x_i})\,\mathrm{w_j}+\mathrm{b_j})\Big) - \Big(\frac{1}{2}\mathrm{x}_i + \mathrm{x}_i(\mathrm{x}_i-1)\big[\sum_{\mathrm{j}=1}^{7}\mathrm{s_j}\,\mathrm{T}(\xi(\mathrm{x_i})\,\mathrm{w_j}+\mathrm{b_j})\big]\Big)^2\Big]^2.$$

Since x in this example is between 0 and 1 and according to theorem , it requires to select $0 < \lambda < 1$, then it is selected $\lambda = 0.6$ and let $\varepsilon = 10^{-6}$. The training set $\vec{\mathrm{x}} = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$,
ξ(x): 0.3 0.39 0.48 0.57 0.66 0.75 0.84 0.93 1.02 1.11 1.2 .
In Figure 3, the analytical and neural solutions found in the training set are shown by the feed forward MNS_1 trained using a grid of evenly spaced points in [0,1]. Then the oral results for MNS_1 , UANN, and train accuracy errors are shown in Table 2 .

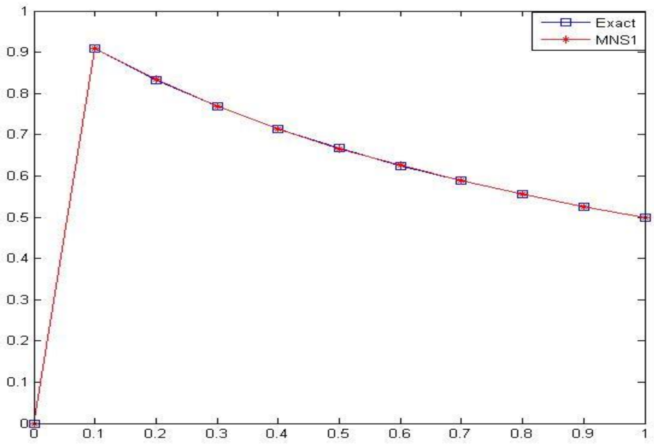


Figure 3: Analytic and MNS_1 of example 2, with $\varepsilon = 10^{-6}$.

Table 2: Analytic, MNS_1 solution and accuracy of the train of example 2,$\varepsilon = 10^{-6}, \lambda = 0.6$.

Input x	Analytic solution $\psi_a(x)$	Solution of MNS_1 $\psi_t(x)$ for training algorithm	Solution of UANN $\psi_t(x)$ for training algorithm	Accuracy of solutions of MNS_1 $E(x) = \lvert\psi_t(x) - \psi_a(x)\rvert$	Accuracy of solutions of UANN $E(x) = \lvert\psi_t(x) - \psi_a(x)\rvert$
0	0.000000000000000	0.000000000000000	0.000000000000000	0	0
0.1	0.909090909090909	0.909090909554321	0.909055432298700	4.63412E-10	3.54768E-05
0.2	0.833333333333333	0.833333337765109	0.833335544983210	4.43178E-09	2.21165E-06
0.3	0.769230769230769	0.769230764432109	0.769230766521986	4.79866E-09	2.70878E-09
0.4	0.714285714285714	0.714285788751908	0.714285788751908	7.44662E-08	7.44662E-08
0.5	0.666666666666667	0.666666640911298	0.666666666543219	2.57554E-08	1.23448E-10
0.6	0.625000000000000	0.625009800543280	0.625008874319860	9.80054E-06	8.87432E-06
0.7	0.588235294117647	0.588235288743098	0.588235288743098	5.37455E-09	5.37455E-09
0.8	0.555555555555556	0.555555886744980	0.555621009654390	3.31189E-07	6.54541E-05
0.9	0.526315789473684	0.526315788875430	0.526315788875430	5.98254E-10	5.98254E-10
1	0.500000000000000	0.500000000000000	0.500005443219000	0	5.44322E-06
The accuracy of the train		Time: 0:00:05, Epoch: 77	Time: 0:00:21, Epoch: 182	MSE=8.74242E-12	MSE=5.14193E-10

7. Conclusions

In this paper, it has been used a new type of update types on the neural networks to solve the singular perturbation problems. This update is to replace the training data with data after compensation with a polynomial of the second degree . After taking several examples and comparing, the results are in the practical side, So the method is characterized by the speed of convergence and reduction error rates and this is clear through the time , epoch and mean squared error in the tables are compared with exact solution and usual artificial neural networks.

Acknowledgements. We appreciate all the people who supported and helped us with essential suggestions and valuable comments for improving this article.

References

[1] A. J. Brown, "*Analysing a Neural Network trained by Backpropagation,*" Neural Computation-Assignment, 2004.

[2] X. H. Jiang, "Asymptotic analysis of a perturbation problem," *Journal of Computational and Applied Mathematics,* vol. 190, no. 1-2, pp. 22-36, 2006.

[3] A. Shikongo, "Numerical Treatment of Non-Linear singular pertubation problems," ph.D. dissertation, University of the Western Cape, 2007.

[4] P. A. Lagerstrom and R. G. Casten, "Basic concepts underlying singular perturbation techniques," *SIAM review,* vol. 14, no. 1, pp. 63-120, 1972.

[5] G. M. Mairaliyev, "Difference method for a singularly perturbed initial value problem," *Turkish Journal of Mathematics,* vol. 22, no. 3, pp. 283-294, 1998.

[6] I. V. Andrianov, J. Awrejcewicz and M. Matyash, "On application of perturbation method with a few perturbation parameters," *Machine Dynamics Problems,* vol. 24, no. 3, pp. 5-10, 2000.

[7] R. Dastres and M. Soori, "Artificial neural network systems," *International Journal of Imaging and Robotics (IJIR),* vol. 21, no. 2, pp. 13-25, 2021.

[8] K. M. Al-Abrahemee, "Modification of high performance training algorithm for solving singular perturbation partial differential equations with cubic convergence," *Journal of Interdisciplinary Mathematics,* vol. 24, no. 7, pp. 2035-2047, 2021.

[9] R. Kareem and K. M. Al-Abrahemee, "Modification artificial neural networks for solving singular perturbation problems," *Journal of Interdisciplinary Mathematics,* vol. 25, no. 5, pp. 1535-1549, 2022.

[10] H. S. Harba, "Hybrid Approach of Prediction Daily Maximum and Minimum Air Temperature for Baghdad City by Used Artificial Neural Network and Simulated Annealing," *Iraqi Journal of Science,* vol. 59, no. 1, p. 591–599, 2018.

[11] A. K. Mohammed and M. K. Dhaidan, "Prediction of Well Logs Data and Estimation of Petrophysical Parameters of Mishrif Formation, Nasiriya Field, South of Iraq Using Artificial Neural Network (ANN)," *Iraqi Journal of Science,* vol. 64, no. 1, p. 253–268, 2023.

[12] Z. J. AL Zirej and H. A. Hadi, "An Artificial Neural Network for Predicting Rate of Penetration in AL- Khasib Formation â€" Ahdeb Oil Field," *Iraqi Journal of Science,* vol. 61, no. 5, p. 1051–1062, 2020.

[13] O. N. Kadhim , F. H. Najjar and K. T. Khudhair, "Detection of COVID-19 in X-Rays by Convolutional Neural Networks," *Iraqi Journal of Science,* vol. 64, no. 4, p. 1963–1974, 2023.

[14] R. K. Dash and P. Daripa, "Analytical and numerical studies of a singularly perturbed Boussinesq equation," *Applied Mathematics and Computation,* vol. 126, no. 1, pp. 1-30, 2002.

[15] J. K. Hunter, "Asymptotic analysis and singular perturbation theory," *Department of Mathematics University of California at Davis,* pp. 1-3, 2004.

[16] T. Valanarasu and N. Ramanujam, "An asymptotic initial value method for second order singular perturbation problems of convection-diffusion type with a discontinuous source term," *Journal of Applied Mathematics and Computing,* vol. 23, pp. 141-152, 2007.

[17] M. Jianzhong, "Some Singular Singularly Perturbed Problem",," M.Sc. Thesis, Calgary, Alberta, 1997.