Eur. Phys. J. Plus (2020) 135:751
https://doi.org/10.1140/epjp/s13360-020-00780-3

THE EUROPEAN
PHYSICAL JOURNAL PLUS

Regular Article

Numerical solution of third-order Emden–Fowler type equations using artificial neural network technique

Akanksha Verma[a], **Manoj Kumar**[b]

Department of Mathematics, Motilal Nehru National Institute of Technology Allahabad, Prayagraj,
U.P. 211004, India

Received: 1 August 2020 / Accepted: 15 September 2020 / Published online: 22 September 2020

Abstract This paper presents the application of artificial neural network technique for solving a class of third-order linear and nonlinear boundary value problems with mixed nonlinear boundary conditions. This technique overcomes the singular behavior of problems and outlines the approximations of high exactness with a vast viable region of convergence. The proposed method has been tested for various examples; acquired outcomes exhibit the effectiveness and robustness of the proposed strategy and are compared with the other existing numerical techniques.

1 Introduction

Singular boundary value problems play a significant role in the field of logical application and emerge from the zone of scientific material science, astronomy, compound physical science, and theoretical physical science leading to the nonlinear marvels, which can demonstrate via the formula of the Lane–Emden–Fowler-type equation

$$r^{-\rho}\frac{d^k}{dr^k}\left(r^{\rho}\frac{d^m}{dr^m}\right)w + f(r)g(w) = 0 \tag{1}$$

where ρ is the shape factor, $f(r)$ and $g(w)$ are some given real-valued functions of r and w, respectively. The Emden–Fowler-type equation (1) for $k = m = 1$ arises in the study of relativistic mechanics, fluid mechanics, population evolution, chemical reactor system, and in the study of pattern formation.

For the particular case, if $f(r) = 1$ and $g(w) = w^n$, then Eq. (1) transforms into the standard form of Lane–Emden-type equation of the first kind, which speaks to the warm history of a circular haze of gas, isothermal gas circle, including aspects of stellar structure [1], and mean-field treatment of a phase transition in basic adsorption and the displaying of groups of cosmic systems. However, for $f(r) = 1$ and $g(w) = e^w$, Eq. (1) turns into the standard Lane–Emden equation of second kind, which models the non-dimensional density distribution $w(r)$ in an isothermal gas sphere.

[a] e-mail: rma1702@mnnit.ac.in (corresponding author)

[b] e-mail: manoj@mnnit.ac.in

To find the Emden–Fowler-type equations of third order, set $k + m = 3, k, m \geq 1$ in Eq. (1). This leads to the following two categorizations:

Case I Substitute $k = 2$ and $m = 1$ in (1), to acquire the first kind of third-order Emden–Fowler-type equations

$$w^{'''} + \frac{2\rho}{r} w^{''} + \frac{\rho(\rho - 1)}{r^2} w^{'} + f(r)g(w) = 0. \tag{2}$$

Here, the singularity appears two times at the points $r = 0$ and $r^2 = 0$, with the shape factors 2ρ and $\rho(\rho - 1)$, respectively. Moreover, the coefficient of $w^{'}$ vanishes at $\rho = 1$, and this results in a shape factor equals 2.

Case II Substitute $k = 1$ and $m = 2$ in (1), to acquire the second kind of third-order Emden–Fowler-type equations

$$w^{'''} + \frac{\rho}{r} w^{''} + f(r)g(w) = 0. \tag{3}$$

Here, the singular point appears only once at $r = 0$, with shape factor k, and this equation does not include the term $w^{'}(r)$.

The initial/boundary conditions for Eqs. (2) and (3) are given as

Type I $w(0) = \alpha, \quad w^{'}(0) = \beta, \quad w^{''}(0) = \gamma$
Type II $w(0) = \alpha_0, \quad w^{'}(0) = \beta_0, \quad w(1) = \gamma_0$
Type III $w(0) = \alpha_1, \quad w^{'}(0) = \beta_1, \quad w^{'}(1) = \gamma_1$

In the ongoing years, many researchers have worked on third-order Emden–Fowler-type equations. The solution of such type singular boundary value problems affected by the singularity occurs at $r = 0$. Many numerical methods have been discussed in order to handle the singularity behavior of such type of boundary value problems, for example, homotopy analysis method [2], homotopy perturbation method [3], variational iteration method [4,5], Adomian decomposition method [6], modified Adomian decomposition method [7,8], Haar wavelet method [9], uniform Haar wavelet resolution technique [10], Cubic and B-Spline methods [11] and nonstandard finite difference method [12].

Caglar et al. [13] presented a fourth-degree B-splines function and a collocation method to discover the solution of third-order nonlinear boundary value problem (BVP) and achieved the first-order accuracy. Further, Khan et al. [14] obtained the solution of third-order nonlinear BVP by utilizing quintic splines with fourth-order accuracy. In 2006, Gao et al. [15] developed a method based on quartic B-splines, to solve the system of third-order BVPs associated with third-order obstacle problems. In order to achieve this, the authors first transform the BVPs to IVPs and then uses quartic splines to solve them. In 2012, Li et al. [16] solved a class of third-order BVPs by utilizing the reproducing kernel method and present the analytic solution in the form of series in reproducing kernel space. Afterward, Dezhbord et al. [17] utilized the reproducing kernel Hilbert space method to beat the complication of the singularity of non-homogeneous, nonlinear third-order singular BVPs.

By using differential transform method (DTM), In 2013, Aruna et al. [18] found the solution of singular nonlinear third-order BVPs by using differential transform method (DTM). Also, Taiwo et al. [19] provide two numerical techniques as iterative decomposition and Bernstein polynomial method. They utilized these techniques to handle the third and fourth-order Emden–Fowler-type equations with mixed boundary conditions. To solve the second-order singular boundary value problems with Neumann and Robin Boundary Conditions, Singh et al. [20] have demonstrated an analytic approach that is based on the variation iteration method

(VIM) and an adjustable parameter h which is known as convergence controlling parameter. They have also discussed the error bound and convergence of the solution. In 2019, Singh [21] studied the Emden–Fowler-type equations with mixed boundary conditions and presented an iterative technique to solve them. This technique is based on Green's function and modified homotopy analysis method (HAM) and produces the series solution. For controlling the convergence of the series solution, this technique contains a flexible parameter which is dictated by minimizing the discrete average residual error. In 2020, Shahni and Singh [22] used the Bernstein collocation method for solving the integral form of Lane–Emden–Fowler-type BVPs. In order to get the solution, the authors first converted the integral equations into the system of nonlinear equations with the help of the Bernstein collocation method and then solve it by using the iterative method.

Nowadays, artificial neural networks method is exceptionally famous among the approximation techniques and has a wide assortment of utilization in various spaces. The solution of nonlinear singular boundary value problems is such an area in which the capacities of feed-forward neural systems have viably used. Classical neural network applications comprise various combinations of perceptrons that together establish the structure called multilayer perceptron. Multilayer perceptron is the original form of artificial neural networks. It is the most commonly used type of neural network in the information investigation analytics field. The basic role of the MLP neural network is to create a model that can tackle complex computational issues from the large sets of data and with multiple variables that are beyond human grasp. In 1994, Lee et al. [23] demonstrated a neural minimization algorithm for solving the differential equations. Further, in 1998, Lagaris et al. [24] solved ordinary and partial differential equations with the help of neural network method by generating a trial solution that fulfills the initial/boundary conditions associated with the differential equation. Afterward, many researchers used ANN method to get the solution of different differential equations [25–28]. Recently, the authors [29] find the solution of Lane–Emden equation with initial conditions by using a multilayer perceptron neural network method and proved the convergence of ANN method with the help of numerical results.

The motivation behind this study is to solve third-order nonlinear singular BVPs by exploiting the MLP ANN model with BFGS quasi-Newton BP algorithm. Sabir et al. [30] designed an integrated intelligent computing paradigm for solving the multi-singular third-order Emden–Fowler-type equations. This paradigm is based on artificial neural networks, genetic algorithms, and the active set algorithm. The authors have also discussed the statistical investigation, accuracy, effectiveness, and convergence of the paradigm.

The employment of ANN technique offers us the following attractive features:

- This technique does not require any linearization process to solve a nonlinear problem.
- Solution given by the ANN method is analytic in nature, i.e., the solution is continuous and differentiable, whereas the solutions, obtained by other numerical technique, are either discrete or have limited differentiability.
- The acquired solution is highly generalized and protects exactness throughout the domain despite the fact that not many focuses have been utilized for obtaining the solution of differential equations.
- We don't have to adjust the technique for various sorts of boundary conditions. A comparative system has followed for illuminating differential equation with a different type of boundary conditions.

  The efficiency and accuracy of the proposed technique for third-order singular BVPs have been illustrated with numerical examples.

  The paper is organized as follows: In Sect. 2, we have presented the MLPNN strategy for solving third-order singular BVPs. Section 3 contains the numerical simulation and results of the problem. In Sect. 4, we discuss the convergence of the error of the numerical method. In Sect. 5, we present the conclusion.

2 Multilayer perceptron neural network technique for third-order singular BVPs

To illustrate the MLPNN technique for the solution of third-order Emden–Fowler-type equations, we have considered the general singular BVP of the following form:

$$r^n w^{'''}(r) = F(r, w(r), w^{'}(r), w^{''}(r)), \tag{4}$$

with certain boundary conditions, where $r \in \mathbb{R}$, $n \in \mathbb{Z}$, $D \subset \mathbb{R}$, D is the domain of definition and $w(r)$ is the solution to be determined. To solve the third-order EFT equation, first, we generate a trial solution by using a feed-forward neural network that satisfies the boundary conditions. After that, we train the neural network to get the solution over the entire domain.

Let $w_t(r, \overrightarrow{p})$ be the trial solution of Eq. (4) with adjustable parameter $\overrightarrow{p}$ and input vector r. Then, the given problem reduces to the subsequent discretize form

$$F\left(r_i, w_t(r_i, \overrightarrow{p}), w_t^{'}(r_i, \overrightarrow{p}), w_t^{''}(r_i, \overrightarrow{p}), w_t^{'''}(r_i, \overrightarrow{p})\right) = 0. \tag{5}$$

Equation (5) is transferred to an unconstrained optimization problem as

$$\min_{\overrightarrow{p}} \sum_{r_i \in D} \frac{1}{2} f\left(r_i, w_t(r_i, \overrightarrow{p}), w_t^{'}(r_i, \overrightarrow{p}), w_t^{''}(r_i, \overrightarrow{p}), w_t^{'''}(r_i, \overrightarrow{p})\right)^2. \tag{6}$$

Here, we consider the trial solution is of the following form [26]

$$w_t(r, \overrightarrow{p}) = A(r) + F\left(r, N(r, \overrightarrow{p})\right), \tag{7}$$

where the first term fulfills the boundary conditions and contains no adjustable parameter. The subsequent term contains an adjustable parameter and not to contribute in boundary conditions. Here, $N(r, \overrightarrow{p})$ is single output feed-forward neural network with parameters $\overrightarrow{p}$ and input r.

The optimization process has considered the training of the feed-forward neural network. Here, we use the back-propagation algorithm, which is an optimization procedure and utilized for the training of the counterfeit neural network. It requires a gradient of network derivative concerning its input.

For the given input vector, the network output $N(r, p)$ can be computed as

$$N(r, p) = \sum_{i=1}^{h} \nu_i \sigma(z_i). \tag{8}$$

where $z_i = \sum_{j=1}^{n} \xi_{ij} r_j + \mu_i$, ξ_{ij} shows the weight from the input node j to hidden node i, ν_i shows the weight from hidden node i to output node, μ_i shows the bias of hidden node i, $\sigma(z)$ is sigmoid activation function.

The gradient of the network output $N(r, p)$ with respect to input vector, we get

$$\frac{\partial^k N}{\partial r^k} = \sum_{i=1}^{h} \nu_i \xi_{ij}^k \sigma_i^{(k)}. \tag{9}$$

where $\sigma_i = \sigma(z_i)$ and $\sigma^{(k)}$ represents the kth-order derivative of activation function.

Generally, the derivative of different orders with respect to different inputs can be composed as

$$\frac{\partial^{m_1}}{\partial {r_1}^{m_1}}\frac{\partial^{m_2}}{\partial {r_2}^{m_2}}\cdots\frac{\partial^{m_n}}{\partial {r_n}^{m_n}}=\sum_{i=1}^{h}\nu_i P_i \sigma_i^{(\Lambda)} \tag{10}$$

where $P_i=\prod_{k=1}^{n}\xi_{ik}^{m_k}$ and $\Lambda=\sum_{i=1}^{n}m_i$

Gradient of network output with respect to its inputs is equivalent to feed-forward neural network $N_g(\overrightarrow{r})$

$$N_g=\sum_{i=1}^{h}\nu_i P_i \sigma_i^{(\Lambda)}. \tag{11}$$

Now the derivative of N_g with respect to network parameters (μ_i, ν_i, ξ_{ij}) can be established as

$$\frac{\partial N_g}{\partial \nu_i}=P_i\sigma_i^{(\Lambda)}. \tag{12}$$

$$\frac{\partial N_g}{\partial \mu_i}=\nu_i P_i\sigma_i^{(\Lambda+1)}. \tag{13}$$

$$\frac{\partial N_g}{\partial \xi_{ij}}=r_j\nu_i P_i\sigma_i^{(\Lambda+1)}+\nu_i m_j\xi_{ij}^{(m_j-1)}\left(\prod_{k=1,k\neq j}\xi_{ik}^{m_k}\right)\sigma_j^{(\Lambda)}. \tag{14}$$

In the wake of assessing all the derivative, the network parameter redesign rule is composed as:

$$\nu_i(t+1)=\nu_i(t)+\delta\frac{\partial N_g}{\partial \nu_i}. \tag{15}$$

$$\mu_i(t+1)=\mu_i(t)+\epsilon\frac{\partial N_g}{\partial \mu_i}. \tag{16}$$

$$\xi_{ij}(t+1)=\xi_{ij}(t)+\zeta\frac{\partial N_g}{\partial \xi_{ij}}. \tag{17}$$

where δ, ϵ, and ζ are learning rates, $i=1,2,\ldots,n$ and $j=1,2,\ldots,h$, where the derivative of the error with respect to the weights has been minimized. At that point, It is a straight forward to utilize any minimization strategy. Therefore, the iterative procedure continuing until the error function comes down to its minimum.

2.1 BFGS quasi-Newton back-propagation algorithm

The Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimization algorithm has been used for the minimization of the unconstrained problem. The training of the MLP has been formulated as a nonlinear unconstrained optimization problem. The objective of a learning process is to find a weight vector that minimizes the difference between the actual output and the desired output. The proposed algorithm is generic and easy to implement in all commonly used gradient-based optimization processes. The development is increasing day by day, so that the vast array of possibilities open to a programmer wishing to implement such a method is daunting. Here, an experiment has been made to framework the basic structure on which such methods are arranged. This subsection describes a set of preferences of technique for approximating the solution of the given problems. Here, we have used the MATLAB optimization toolbox to tackle this problem and, minimized the error function with *fminunc*, for more details [31].

3 Numerical results

In this section, we have handled the following three problems of third-order Emden–Fowler-type equations by utilizing the MLPNN technique. The problems have been solved by utilizing the proposed neural network technique by taking 5 neurons in a hidden layer, which conducts about 15 obscure versatile parameters. Since the exactness of approximation relies upon the number of neurons in the hidden layer and doesn't rely upon the number of hidden layers. Here, we have used the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton back-propagation algorithm for the minimization of connection weights between two layers. PC programs have been executed in MATLAB 16.0. In each problem, we have used the objective function minimizer *fminunc*.

Problem 3.1 *Consider the third-order Emden–Fowler-type equation* [6,32]

$$w'''(r) + \frac{6}{r}w''(r) + \frac{6}{r^2}w'(r) = 6(r^6 + 2r^3 + 10)e^{-3w(r)}, \quad 0 \le r \le 1 \tag{18}$$

subject to $w(0) = 0,\ w'(0) = 0,\ w''(0) = 0.$

The analytic solution for Problem 3.1 is $\ln(1 + r^3)$. The problem defined in Eq. (18) is first kind third-order Emden–Fowler-type equation.

We can write the trial solution for Problem 3.1 as

$$w_t(r, p) = r^3 N(r, p). \tag{19}$$

We have trained the network for ten equidistant points in domain [0, 1], by taking 5 sigmoid unit in hidden layer. In Table 1, we have presented the ideal estimation of weights and biases. The problem is solved in 173 iterations and acquired minimum objective function value is 2.9607×10^{-8}. Table 2 shows comparison among numerical solution acquired by VIM method [4], Quartic B-Spline method at $h = 0.025$ [33] and New Cubic B-Spline method at $h = 0.025$ [32] and MLPNN method. Figure 1 depicts the resemblance between the exact solution and MLPNN solution and error between the exact solution and MLPNN solution is plotted in Fig. 2.

Problem 3.2 *Consider the following third-order IVP* [10,32]

$$w'''(r) + \frac{2}{r}w''(r) = \frac{9}{8}(r^6 + 8)w^{-5}(r), \quad 0 \le r \le 1 \tag{20}$$

subject to $w(0) = 1,\ w'(0) = 0,\ w''(0) = 0.$

Table 1 The ideal estimation of weight and biases (Problem 3.1)

i	1	2	3	4	5
μ_i	0.098866	0.465916	1.805384	−0.733968	0.105327
ν_i	−1.326535	−0.638925	0.709974	1.796852	0.955831
ξ_i	−1.242562	0.610944	−0.290230	−0.065530	−0.470905

Table 2 Comparison of numerical results for Problem 3.1

r	MLPNN	Exact Solution	VIM [4]	QBSM [33] $h = 1/40$	New Cubic [32] B-Spline $h = 1/40$
0	0	0	0	0	0
0.1	0.001013	0.001000	0.001000	0.000999	0.000999
0.2	0.007986	0.007968	0.007968	0.007968	0.007968
0.3	0.026597	0.026642	0.026642	0.026641	0.026642
0.4	0.061910	0.062035	0.062035	0.062032	0.062035
0.5	0.117634	0.117783	0.117778	0.117777	0.117783
0.6	0.195444	0.195567	0.195487	0.195558	0.195567
0.7	0.294813	0.294906	0.294166	0.294896	0.294906
0.8	0.413361	0.413433	0.408487	0.413426	0.413434
0.9	0.547508	0.547543	0.521812	0.547544	0.547544
1.0	0.693171	0.693147	0.583333	0.693160	0.693147

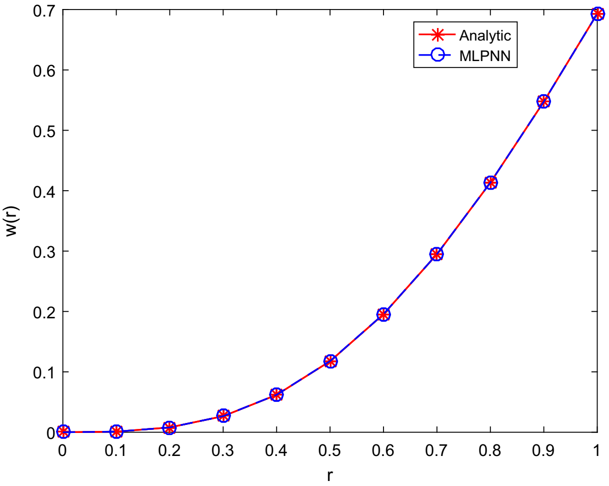

Fig. 1 Plot of exact solution and MLPNN solution (Problem 3.1)

The analytic solution for Problem 3.2 is $\sqrt{1+r^3}$. The problem defined in (20) is the second kind third-order Emden–Fowler-type equation.

We can write the trial solution for Problem 3.2 as

$$w_t(r, p) = 1 + r^3 N(r, p). \tag{21}$$

We have trained the network for ten equidistant points in domain [0, 1], by taking 5 sigmoid unit in hidden layer. In Table 3, we have presented the ideal estimations of weights and biases. The problem is solved in 101 iterations and acquired minimum objective function value is 3.0569×10^{-10}. Table 4 shows comparison among numerical solution acquired by VIM method [4], Quartic B-Spline method at $h = 0.025$ [33], New Cubic B-Spline method at $h = 0.025$ [32] and MLPNN method. Figure 3 depicts the resemblance between the analytic

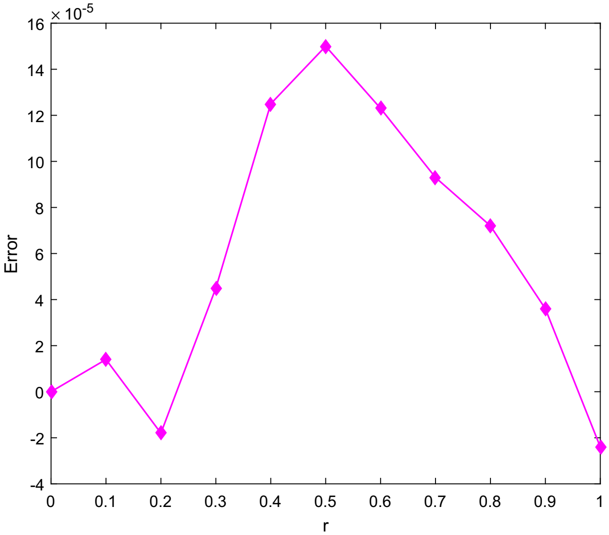

Fig. 2 Plot of error between exact solution and MLPNN solution (Problem 3.1)

Table 3 The ideal estimations of weight and biases (Problem 3.2)

i	1	2	3	4	5
μ_i	0.640337	0.699067	1.191231	0.632595	−0.512441
ν_i	−0.404837	0.611317	−0.038435	1.099170	0.584731
ξ_i	0.921925	1.275595	0.059691	0.935976	0.986927

solution and MLPNN solution and error between the exact solution and MLPNN solution is plotted in Fig. 4.

Problem 3.3 *Consider the third-order Emden–Fowler-type BVP* [10,32]

$$w^{'''}(r) - \frac{2}{r}w^{''}(r) = w^3(r) - r^9e^{3r} + r^3e^r + 7r^2e^r + 6re^r - 6e^r, \quad 0 \le r \le 1 \qquad (22)$$

subject to $w(0) = 0$, $w^{'}(0) = 0$, $w^{'}(1) = 4e$.

The analytic solution for Problem 3.3 is r^3e^r. The problem defined in (3.3) is the second kind third-order Emden–Fowler-type equation.

We can write the trial solution for Problem 3.3 as

$$w_t(r, p) = r^4e + r^2(r-1)^2N(r, p). \qquad (23)$$

We have trained the network for ten equidistant points in domain [0, 1], by taking 5 sigmoid units in the hidden layer. In Table 5, we have presented the ideal estimations of weights and biases. The problem is solved in 273 iterations and the acquired minimum objective function value is 3.0618×10^{-9}. Table 6 shows comparison among numerical solution acquired by MADM [8], DTM [18], Quartic B-Spline method at $h = 0.025$ [33], New Cubic B-Spline method at $h = 0.025$ [32] and MLPNN method. Figure 5 depicts the resemblance between

Table 4 Comparison of numerical results for Problem 3.2

r	MLPNN	Exact solution	VIM [4]	QBSM [33] $h = 1/40$	New Cubic [32] B-Spline $h = 1/40$
0	1	1	1	1	1
0.1	1.000486	1.000500	1.000500	1.000500	1.000500
0.2	1.003957	1.003992	1.003992	1.003992	1.003992
0.3	1.013478	1.013410	1.013410	1.013409	1.013410
0.4	1.032031	1.031504	1.031504	1.031502	1.031504
0.5	1.062344	1.060660	1.060659	1.060655	1.060660
0.6	1.106744	1.102724	1.102713	1.102715	1.102724
0.7	1.167028	1.158879	1.158775	1.158866	1.158879
0.8	1.244348	1.229634	1.228936	1.229617	1.229634
0.9	1.339113	1.314914	1.311251	1.314894	1.314915
1.0	1.450907	1.414214	1.398438	1.414191	1.414214

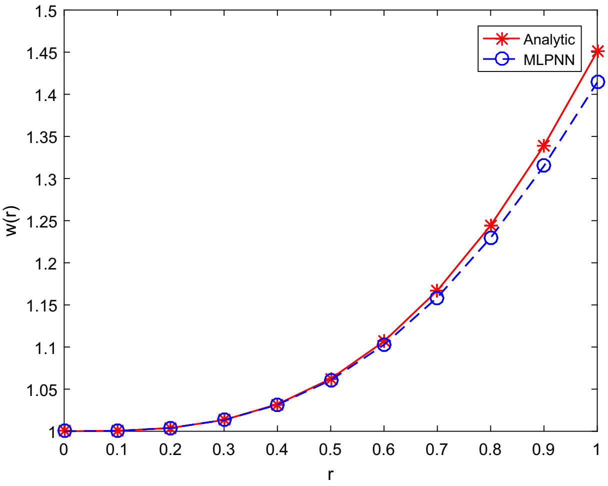

Fig. 3 Plot of exact solution and MLPNN solution (Problem 3.2)

the exact solution and MLPNN solution and error between the exact solution and MLPNN solution is plotted in Fig. 6.

Problem 3.4 *Consider the third-order Emden–Fowler-type IVP* [30,34]

$$w^{'''}(r) + \frac{4}{r}w^{''}(r) - (r^6 + 10t^3 + 10)w(r) = 0, \quad 0 \le r \le 1 \tag{24}$$

subject to $w(0) = 1$, $w^{'}(0) = 0$, $w^{''}(0) = 0$.

The exact solution for Problem 3.4 is $e^{\frac{r^3}{3}}$. This problem is the second kind third-order Emden–Fowler-type equation.

We can write the trial solution of Problem 3.4 as

$$w_t(r, p) = 1 + r^3 N(r, p). \tag{25}$$

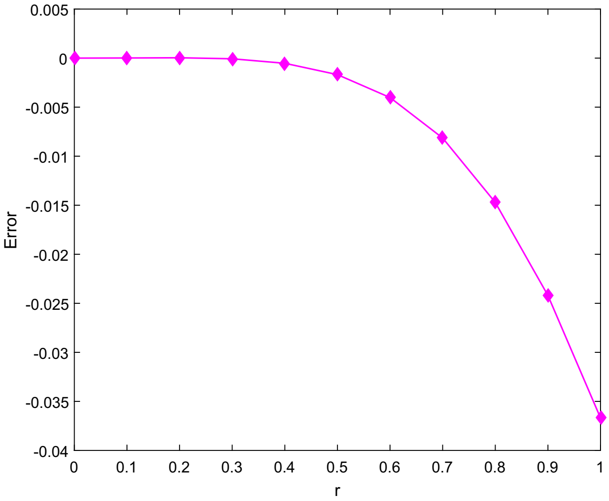

Fig. 4 Plot of error between exact solution and MLPNN solution (Problem 3.2)

Table 5 The ideal estimations of weight and biases (Problem 3.3)

i	1	2	3	4	5
μ_i	0.865203	0.901608	0.749090	1.433634	1.518882
ν_i	−0.045855	−1.224268	0.302917	0.700943	1.014284
ξ_i	0.222630	1.341001	0.105585	0.658269	0.563485

Table 6 Comparison of numerical results for Problem 3.3

r	MLPNN	Exact solution	MADM [8]	DTM [18]	QBSM [33] $h = 1/50$	New Cubic [32] B-Spline $h = 1/50$
0.0	0	0	0	0	0	0
0.1	0.000574	0.001105	0.001103	0.001105	0.001105	0.001105
0.2	0.008769	0.009771	0.009732	0.009768	0.009768	0.009771
0.3	0.034552	0.036446	0.036248	0.036431	0.036435	0.036446
0.4	0.091173	0.095477	0.094852	0.095429	0.095453	0.095477
0.5	0.197852	0.206090	0.204564	0.205973	0.206048	0.206090
0.6	0.381389	0.393578	0.390412	0.393334	0.393515	0.393577
0.7	0.676964	0.690717	0.684849	0.690265	0.690637	0.690717
0.8	1.128451	1.139477	1.129458	1.138703	1.139393	1.139476
0.9	1.788459	1.793051	1.776975	1.791806	1.792987	1.793050
1.0	2.718282	2.718282	2.693707	2.716368	2.718282	2.718282

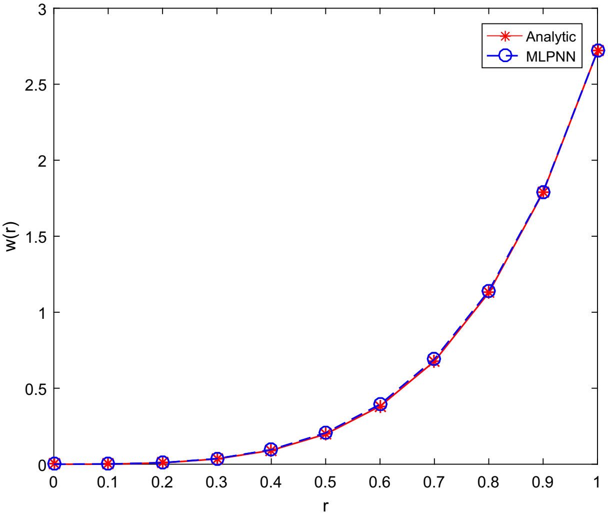

Fig. 5 Plot of exact solution and MLPNN solution (Problem 3.3)

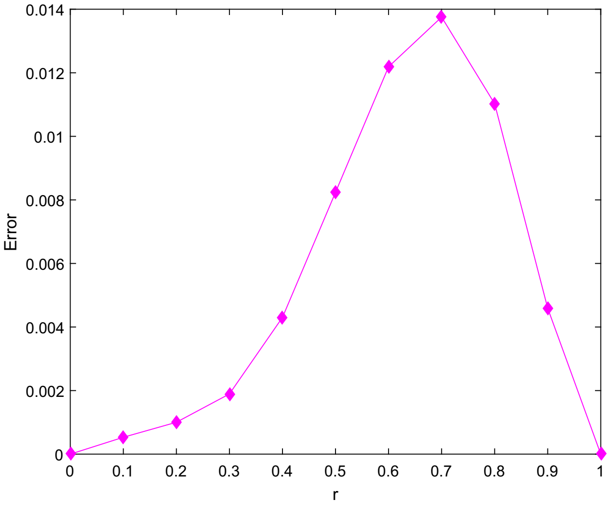

Fig. 6 Plot of error between exact solution and MLPNN solution (Problem 3.3)

We have trained the network for ten equidistant points in domain [0, 1], by taking 5 sigmoid units in the hidden layer. In Table 7, we have presented the ideal estimation of weights and biases. The problem is solved in 273 iterations and the acquired minimum objective function value is 6.8590×10^{-7}. Table 8 shows the comparison of the MLPNN solution with exact solution. Figure 7 depicts the resemblance between the exact solution and MLPNN solution and error between the exact solution and MLPNN solution is plotted in Fig. 8.

Table 7 The ideal estimations of weight and biases (Problem 3.4)

i	1	2	3	4	5
μ_i	−0.334393	1.524535	−0.149362	0.53811	0.426934
ν_i	0.867063	0.765903	−1.290836	1.345371	−0.818204
ξ_i	3.047906	−3.141277	0.529794	1.570969	0.492431

Table 8 Comparison of numerical results for Problem 3.4

r	MLPNN	Exact solution	Absolute error (MLPNN)
0.0	1	1	0
0.1	1.000334	1.000333	1.0×10^{-6}
0.2	1.002672	1.002670	2.0×10^{-6}
0.3	1.009042	1.009041	1.0×10^{-6}
0.4	1.021562	1.021563	1.0×10^{-6}
0.5	1.042546	1.042547	1.0×10^{-6}
0.6	1.074655	1.074655	1.0×10^{-6}
0.7	1.121124	1.121126	2.0×10^{-6}
0.8	1.186090	1.186095	5.0×10^{-6}
0.9	1.275065	1.275069	3.0×10^{-6}
1.0	1.395620	1.395612	7.0×10^{-6}

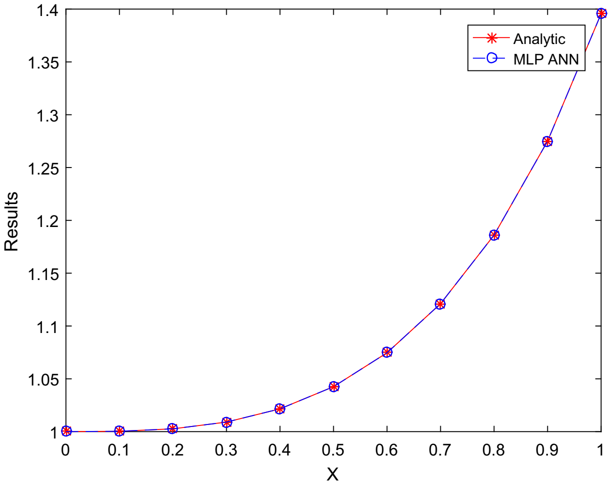

Fig. 7 Plot of exact solution and MLPNN solution (Problem 3.4)

4 Convergence analysis

From the error analysis, we conclude that the MLPNN technique solutions accuracy relies on the number of neurons in the hidden layer as we increase the hidden units [34], the multifaceted nature and exactness increase. The detailed convergence analysis for the proposed technique has been established by absolute error, plotted graphs, and time required for the solution.

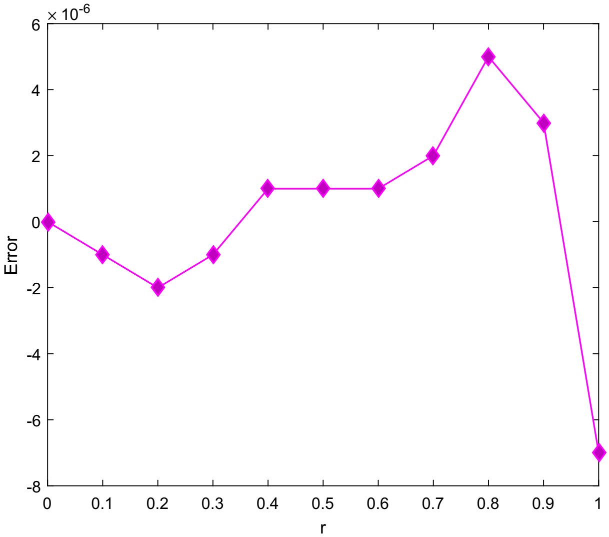

Fig. 8 Plot of error between exact solution and MLPNN solution (Problem 3.4)

Table 9 Absolute errors and time required for problems

Problem no.	Max. absolute error	Min. absolute error	Computational time
1	1.23×10^{-4}	1.4×10^{-5}	0.834991 s
2	3.66×10^{-2}	1.3×10^{-5}	0.481167 s
3	1.37×10^{-2}	5.31×10^{-4}	0.050931 s
4	7.0×10^{-6}	1.0×10^{-6}	0.589943 s

Moreover, the technique is straightforwardly applicable to problems with singular points. The error calculated in Table 9 indicates the proposed method is convergent.

5 Conclusion

The theory of the singular boundary value problems finds its indispensable presence in many of the natural/physical processes, for example, oxygen diffusion in spherical cells [35], heat conduction in the human head [36], and in shallow membrane cap [37]. In this article, we have solved the third-order Emden–Fowler-type equations of two kinds using a multilayer perceptron neural network technique. The parameters of MLPNN have been minimized by using the back-propagation algorithm and the quasi-Newton method. The accuracy and convergence of the present method are analyzed through the outcomes of the problem. Numerical outcomes show that the MLPNN method is comparable to the reported results of other approximation techniques like VIM [4], QBSM [33], MADM [8], DTM [18], and Cubic B-Spline method [32]. However, the precision of the New Cubic B-Spline method and uniform Haar wavelet resolution technique is better than the MPLNN approach. But it relies on optimum spatial and temporal discretization value. The computational outcomes show that the present MLPNN technique effectively consolidates exactness and efficiency doesn't require any lattice discretization. Indeed, even less number of training points inside the domain can accurately illuminate the third-order Emden–Fowler-type equation which minimizes the computational time also. This is a novel approach for the third-order Emden–Fowler-type equation. So we can say that the procedure is incredible and relevant for a wide extent of singular BVPs emerging in any engineering and science applications.

Acknowledgements The authors are thankful to the National Board of Higher Mathematics (NBHM), Government of India for providing financial support to bring out this work through its project sanctioned Order No. 02011/-25/2019/R&D-II/3889. We express our sincere thanks to editor in chief, editor, and reviewers for their valuable suggestions to revise this manuscript.

References

1. R. Emden, (Leipzig, Teubner, 1907)
2. O.P. Singh, R.K. Pandey, V.K. Singh, Comput. Phys. Commun. **180**, 2009 (2009)
3. M.S.H. Chowdhury, I. Hashim, Nonlinear Anal. Real World Appl. **10**, 1 (2009)
4. A.M. Wazwaz, Appl. Math. Inform. Sci. **9**, 5 (2015)
5. A.M. Wazwaz, J. Math. Chem. **55**, 3 (2016)
6. A.M. Wazwaz, R. Rach, L. Bougoffa, J.S. Duan, Comput. Model. Eng. Sci. **100**, 6 (2014)
7. G. Adomian, R. Rach, Nonlinear Anal. Theory Methods Appl. **23**, 5 (1994)
8. Y.Q. Hasan, L.M. Jhu, Commun. Nonlinear Sci. Numer. Simul. **14**, 2009 (2009)
9. R. Singh, H. Garg, V. Guleria, J. Comput. Appl. Math. **346**, 2019 (2019)
10. M. Singh, K. Swati, A. Singh, K. Verma, J. Comput. Appl. Math. **376**, 2020 (2020)
11. P. Roul, K. Thula, Int J. Comput. Math. **96**, 2017 (2017)
12. A.K. Verma, S. Kayenat, J. Math. Chem. **56**, 2018 (2018)
13. H.N. Caglar, S.H. Caglar, E.H. Twizell, Int. J. Comput. Math. **71**, 1998 (1998)
14. A. Khan, T. Aziz, Appl. Math. Comput. **137**, 253 (2003)
15. F. Gao, C.M. Chi, Appl. Math. Comput. **180**, 270 (2006)
16. Z. Li, Y. Wang, F. Tan, Abstr. Appl. Anal. (2012)
17. A. Dezhbord, T. Lotfi, K. Mahdiani, Adv. Differ. Equ. **161**, 2018 (2018)
18. K. Aruna, A.S.V. Ravi Kantha, Int. J. Pure Appl. Math. **84**, 4 (2013)
19. O.A. Taiwo, M.O. Hassan, Br. J. Math. Comput. Sci. **9**, 6 (2015)
20. R. Singh, N. Das, J. Kumar, Eur. Phys. J. Plus. **132**, 251 (2017)
21. R. Singh, Eur. Phys. J. Plus **134**, 583 (2019)
22. J. Shahni, R. Singh, Eur. Phys. J. Plus **135**, 475 (2020)
23. H. Lee, I.S. Kang, J. Comput. Phys. **91**, 110 (1990)
24. I.E. Lagaris, A. Likas, D.I. Fotiadis, IEEE Trans. Neural Netw. **9**, 987 (1998)
25. K.S. Mcfall, J.R. Mahan, IEEE Trans. Neural Netw. **20**, 2009 (2009)
26. M. Kumar, N. Yadav, Comput. Math. Appl. **62**, 2011 (2011)
27. M. Kumar, N. Yadav, J. Franklin I. **350**, 10 (2013)
28. S. Mall, S. Chakraverty, Neurocomputing **149**, 2015 (2015)
29. A. Verma, M. Kumar, Int. J. Appl. Comput. Math. **5**, 141 (2019)
30. Z. Sabir, M.Umar, J. L. G. Guirao, M. Shoaib,M. A. Z. Raja, Neural Comput. Appl. (2020)
31. MATLAB Optimization Toolbox, MATLAB R2016a, The MathWorks, Natick, MA, USA
32. M.K. Iqbal, M. Abbas, I. Wasim, Appl. Math. Comput. **331**, 2018 (2018)
33. H.K. Mishra, S. Saini, Am. J. Numer. Anal. **3**, 1 (2015)
34. A. Jafarian, Int. J. Ind. Math. **7**, 2015 (2015)
35. D. McElwain, J. Theor. Biol. **71**, 1978 (1978)
36. B. Gray, J. Theor. Biol. **82**, 3 (1980)
37. I. Rachnková, O. Koch, G. Pulverer, E. Weinmuller, J. Math. Anal. Appl. **332**, 1 (2007)