The First International Conference on
Machine Learning and Knowledge Discovery (MLKD 2024)
Amirkabir University of Technology, December 18-19, 2024

The Neural Network Method for Solving Lane–Emden Equation with Chebyshev Polynomials of Second Kind

Marzieh Pourbabaee*

Abstract

The objective of this paper is to present two effective computational schemes for solving Lane–Emden-type equations using an artificial neural network. The specified neural network consists of three layers: the input layer, the hidden layer, and the output layer. For the activation functions of the hidden layer, we consider Chebyshev polynomials of the second kind. Also, we consider another activation function for the output layer. Finally, for train, this neural network, collocation method and classical optimization method are applied. The applicability and accuracy of the expressed technique are investigated in three illustrative examples.

Keywords: Neural network, Lane–Emden equation, and Chebyshev polynomials of second kind

1 Introduction

The theory of singular boundary value problems has become an important area of investigation in the past three decades (see [1, 2, 3, 4, 5]). One of the equations describing this type is the Lane-Emden equation. In astrophysics, the Lane–Emden equation represents a dimensionless version of Poisson's equation, which describes the gravitational potential of a Newtonian fluid that is both self-gravitating and spherically symmetric, with polytropic characteristics. This equation is named after the astrophysicists Jonathan Homer Lane and Robert Emden. Nonlinear singular initial value problems (IVPs) in ordinary differential equations (ODEs) indeed arise in various scientific and engineering applications, reflecting complex physical phenomena. In particular, these ODEs can exhibit behaviors that are sensitive to initial conditions and can exhibit singularities that challenge traditional analytical and numerical methods. Some examples of applications includethe polytropic theory of stars, thermodynamics, energy transport models, stellar structure, radioactive cooling, and modeling clusters of galaxies. These problems are often difficult to solve analytically due to the nonlinearity and singular points, thus necessitating numerical methods and advanced mathematical techniques for analysis and solution. Numerical techniques like adomian decomposition methods [6], hybrid functions[7], Lagrangian interpolation method[8], Jacobi matrix method[9] and etc. can be particularly useful for obtaining approximate solutions in these contexts. Typically, the Lane–Emden type equations are expressed as:

$$u^{''}(x) + \frac{\beta}{x} + G(x, u(t)) = H(x), \tag{1}$$

with the initial conditions

$$u(0) = \alpha_0, \quad u^{'}(0) = \alpha_1, \tag{2}$$

where, $H(x)$ and $G(x, u(x))$ are some given functions and $\beta, \alpha_0, \alpha_1$ are real constants. If we consider $\beta = 2, G(x, u(x)) = u^n(x), H(x) = 0$ and $\alpha_0 = 1, \ \alpha_1 = 0$ the standard Lane–Emden equation obtain as follow:

$$u^{''}(x) + \frac{2}{x}u^{'}(x) + u^n(x) = 0, \tag{3}$$

Also, if consider $\beta = 2, G(x, u(x)) = exp(u(x)), H(x) = 0$ and $\alpha_0 = 0, \ \alpha_1 = 0$ the isothermal gas spheres equation obtain as follow:

$$u^{''}(x) + \frac{2}{x}u^{'}(x) + exp(u(x)) = 0, \tag{4}$$

Moreover, if we consider $\beta = 2, G(x, u(x)) = sinh(u(x)), H(x) = 0$ and $\alpha_0 = 1, \ \alpha_1 = 0$, one of the Lane–Emden type equations will be obtain as:

$$u^{''}(x) + \frac{2}{x}u^{'}(x) + sinh(u(x)) = 0, \tag{5}$$

There are several alternative analytical and numerical techniques for obtaining solutions of Lane–Emden-type nonlinear equations. Some of these methods are the Homotopy analysis method [10], Bernstein polynomials method [11], Legendre wavelets method[12], B-spline expansion method [13] and etc. Machine learning, particularly deep learning methods that utilize Neural Networks (NNs), are extensively applied to address challenges across diverse fields such as computer vision, language processing, game theory, and more ([14] and related references). Today, the use of artificial intelligence to solve different types of differential equations has become increasingly common. Malek and Beidokhti

*Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Kashan, Kashan 87317-53153, Iran, m.pourbabaee@kashanu.ac.ir

Shekari in [15] developed a hybrid scheme that combines artificial neural networks (ANNs) with optimization methods to solve higher-order differential equations. Jianyu et al. [16] applied a radial basis neural network for the numerical solution of the elliptical partial differential equation. Masood et al. [17] designed a mexican hat wavelets-based neural network for solving the nonlinear Bratu-type equation.

Given their capacity for universal approximation, neural networks can be an effective method for finding approximate solutions to various initial and boundary value problems. In addition, using neural networks to approximate solutions can offer several advantages over traditional numerical methods. For example, we can compute a differential function without the need for extensive formal numerical calculations, and the computational complexity remains constant regardless of the number of sample points. At any arbitrary point, even between the training points, the solution can be obtained seamlessly. Conversely, the neural network method can effectively tackle both linear and nonlinear ordinary and partial differential equations. Additionally, some researchers have utilized the ANN method to find approximate solutions to the Lane-Emden equation such as Legendre neural network [18], Chebyshev neural network [19], Neural network method based on local search algorithm[20], Fractional orthogonal neural network [21].

2 Chebyshev polynomials of second kind.

The Chebyshev polynomials are regarded as one of the most useful families of polynomials in numerical analysis. They are particularly well-suited for applications such as polynomial approximation, integral and differential equations, and spectral methods for solving partial differential equations [22, 23]. Chebyshev polynomials of the second kind of degree m are defined on the interval $[-1, 1]$ as follows [24]:

$$U_m(x) = \frac{\sin(m+1)\theta}{\sin\theta}, \tag{6}$$

where $\theta = \arccos(x)$. This expression shows that the Chebyshev polynomials of the second kind can be represented using the sine function, emphasizing their periodic properties and connections to trigonometric functions. These polynomials can also be recursively generated starting with $U_0(x) = 1,\ U_1(x) = 2x$ and using the recurrence relation:

$$U_m(x) = 2xU_{m-1}(x) - U_{m-2}(x), \quad m = 2, 3, \ldots, \tag{7}$$

The polynomials $U_m(x)$ exhibit a variety of properties, including orthogonality with respect to the weight function $\sqrt{1-x^2}$ on the interval $[-1, 1]$. To use the Chebyshev polynomials of the second kind on the interval $(0, 1)$, we can apply a change of variable. This involves shifting the variable x from the interval $[-1, 1]$ to $(0, 1)$ using the transformation $2x - 1$. Thus, the shifted second kind Chebyshev polynomial $U^*(x)$ defined on the interval $(0, 1)$ can be expressed as:

$$U^*_m(x) = (4x-2)U^*_{m-1}(x) - U^*_{m-2}(x), \quad m = 2, 3, \ldots, \tag{8}$$

with the initial conditions

$$U^*_0(x) = 1,\ U^*_1(x) = 4x - 2.$$

The analytical expression for the shifted Chebyshev polynomials $U^*_m(x)$ of degree m is provided by

$$U^*_m(x) = \sum_{k=0}^{m+1} k(-1)^{m+1-k} \frac{(m+k)!2^{2k-1}}{(m+1-k)!(2k)!} x^{k-1}. \tag{9}$$

The shifted second kind of Chebyshev polynomials $U^*_m(x)$ retain the orthogonality property over the interval $(0, 1)$ with respect to the weight function $w(x) = \sqrt{x - x^2}$

$$\int_0^1 U^*_m(x)U^*_n(x)w(x) = \lambda\delta_{mn}, \tag{10}$$

where and δ_{mn} is the Kronecker function and $\lambda = \frac{1}{8}\pi$.

3 Structure of Chebyshev polynomials neural network

For solving the presented problem in Equations (1)-(2), we consider the second kind of Chebyshev polynomials neural network (SCPNN). The structure of SCPNN has three layers.

- In a typical neural network architecture, the first layer is often referred to as the input layer which consists of a single node labeled x. This node has a set of data represented as $\{x_1, x_2, \ldots, x_h\}$, where h is number of data.
- The second layer is the hidden layer of the network, in which we choose a class of orthogonal functions as activation function ($\mathbb{AF}$). We have used Chebyshev polynomials of the second kind as activation functions.
- The third layer is the network's output layer. Its input is a linear combination of Chebyshev polynomials of the second kind from the second layer, and its output is the result of applying an activation function to the input.

Therefore, the output of the SCPNN with input data x and parameter B is as

$$N(x, B) = \mathbb{AF}(R), \tag{11}$$

R represents a linear combination of Chebyshev polynomials of the second kind and is represented as

$$R = \sum_{k=0}^{M} b_k U_k^*(x) = B^T \Pi(x), \tag{12}$$

where $B = [b_0, b_1, \ldots, b_M]$ and $\Pi(x) = [U_0^*(x), U_1^*(x), \ldots, U_M^*(x)]$. Training data sets $\{x_1, x_2, \ldots, x_h\}$ can be selected through various methods, including equidistant points and the roots of $M+1$ th the Legendre polynomial, among others. In this paper, we consider the roots of $M+1$th Chebyshev polynomials of the second kind as the training data set.

4 Application of the methods to the Lane–Emden equation

This section discusses the application of Chebyshev polynomials of the second kind in a neural network framework to obtain numerical solutions for the general form of the Lane–Emden equation, as outlined in equations (1) and (2). At first, to address solving the problem (1) and (2) by the numerical approximation methods consider [25]

$$u(x) \simeq \hat{u}(x) = \alpha_0 + \alpha_1 x + x^2 N(x, B), \tag{13}$$

It seems that $\hat{u}(x)$ satisfies certain initial conditions. Also the approximate solution should be satisfied in Equation (1). So, we obtain

$$Res(x, B) = \hat{u}^{''}(x) + \frac{\beta}{x}\hat{u}^{'}(x) + G(x, \hat{u}(t)) - H(x). \tag{14}$$

4.1 Approach1: Chebyshev polynomials of the second kind neural network-collocation (SCPNN-C)

In this technique, we collocate the Equation (14) at $M+1$ zeros of Chebyshev polynomials of the second kind as:

$$Res(x_\ell, B) = \hat{u}^{''}(x_\ell) + \frac{\beta}{x_\ell}\hat{u}^{'}(x_\ell) + G(x_\ell, \hat{u}(x_\ell)) - H(x_\ell), \tag{15}$$

where $\ell = 1, 2, \ldots, M+1$. Now by using fsolve command in Maple software we can obtain B.

4.2 Approach2: Chebyshev polynomials of the second kind neural network-optimization (SCPNN-O)

In this method collocate Equation (14) in every training point $\{x_1, x_2, \ldots, x_h\}$. So, we obtain

$$Res(x_k, B) = \hat{u}^{''}(x_k) + \frac{\beta}{x_k}\hat{u}^{'}(x_k) + G(x_k, \hat{u}(x_k)) - H(x_k). \tag{16}$$

Then we obtain the followin optimization relation as:

$$B^* = \min_B \frac{1}{2}\sum_{j=1}^{h} Res^2(x_j, B). \tag{17}$$

Moreover, Equation (17) can be formed as an unconstrained parametric optimization problem. In an unconstrained optimization problem, we aim to find the parameters (which may also be variables dependent on another parameter) that minimize or maximize an objective function without any restrictions or constraints on those parameters. Now, we should implement the structured approach to solving for the vector B that minimizes the objective function B^*. The necessary conditions for this aim are as

$$\frac{\partial B^*}{\partial b_i}, \quad i = 1, \ldots, h. \tag{18}$$

Now, for finding the unknown vector B we should apply Newton's iterative approach. After that the approximation solution in Equation (13) can be obtained.

5 Numerical examples

In this section, we consider three problems in (3)-(5). For these problems, we consider the active function $\frac{R}{1+|R|}$ for approach 1 and R for approach 2.

Example 1 *In the first example, we consider the standard Lane–Emden equation as introduced in Equation* (3). *The initial condition for this problem as*

$$u(0) = \alpha_0 = 1 \text{ and } u^{'}(0) = \alpha_1 = 0.$$

In Equation (3), *n is a constant. In this example, we consider three cases:*

Case 1: if $n = 0$, the exact solution is $u(x) = 1 - \frac{x^2}{6}$,

Case 2: if $n = 1$, the exact solution is $u(x) = \frac{\sin(x)}{x}$,

Case 3: if $n = 5$, the exact solution is $u(x) = \left(1 + \frac{x^2}{3}\right)^{-\frac{1}{2}}$.

Figure 1 displays the graph of absolute error for both methods with $M = 2$ for Case 1. Also, we report the maximum absolute error with $M = 2$ by applying Method 1 and 2 for Case 2. Finally, the graph of absolute error with $M = 2$ by using Method 1 and $M = 3$ by utilizing Method 2 for Case 3 is displayed in Figure 2.

Example 2 *The equation for an isothermal gas sphere, is expressed as follows:*

$$u^{''}(x) + \frac{2}{x}u^{'}(x) + e^{u(x)} = 0, \tag{19}$$

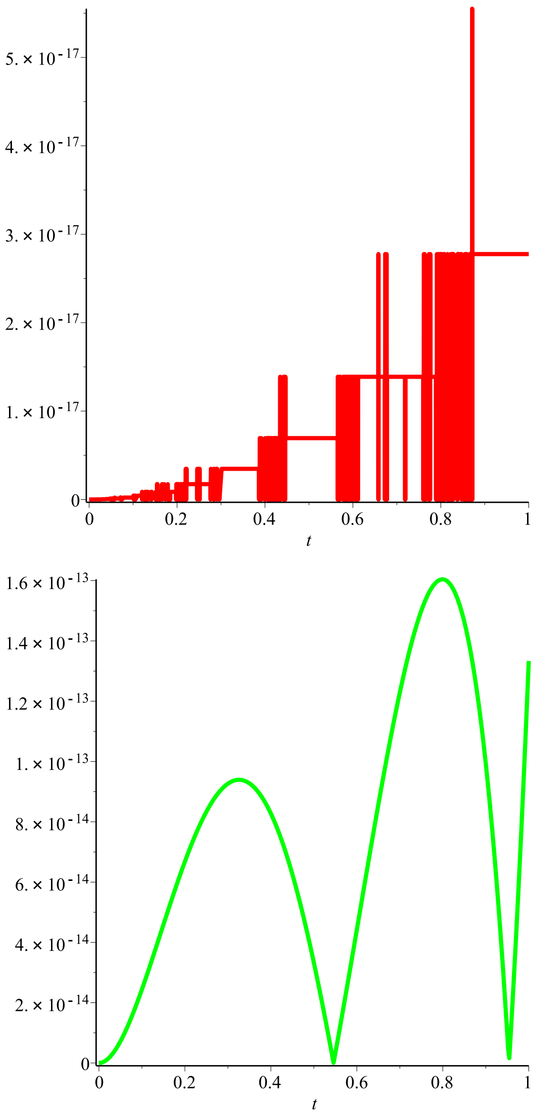

Figure 1: The absolute errors for $M = 2$ for approach 1 (above) and approach 2 (below) for case 1 in Example 1.

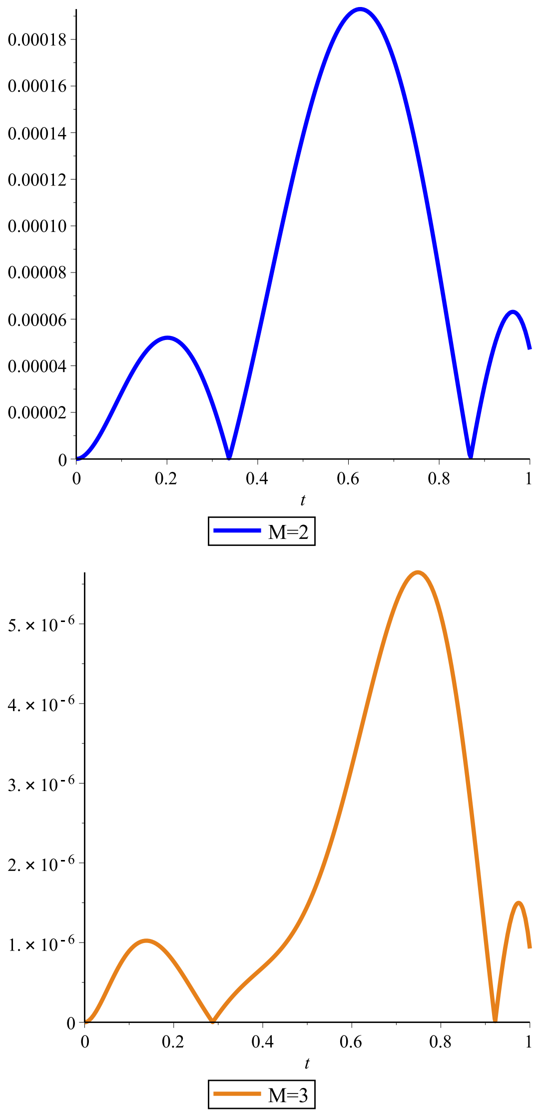

Figure 2: The absolute errors for $M = 2$ for approach 1 (above) and $M = 3$ for approach 2 (below) for case 3 in Example 1.

Table 1: Maximum absolute error with $M = 2$ for case 2 in Example 1.

t	Approach 1	Approach 2
0.1	1.01×10^{-6}	7.36×10^{-7}
0.2	1.88×10^{-6}	1.36×10^{-6}
0.3	8.58×10^{-7}	6.20×10^{-7}
0.4	2.05×10^{-6}	1.48×10^{-6}
0.5	5.54×10^{-6}	4.02×10^{-6}
0.6	7.74×10^{-6}	5.61×10^{-6}
0.7	7.00×10^{-6}	5.07×10^{-6}
0.8	3.00×10^{-6}	2.18×10^{-6}
0.9	2.05×10^{-6}	1.47×10^{-6}

where $u(0) = 0$ *and* $u^{'}(0) = 0$ *can be consider for initial conditions. The solution of Equation* (19) *was approximated in [26] as*

$$u(x) \simeq \frac{-x^2}{6} + \frac{x^4}{5 \times 4!} - \frac{8x^6}{21 \times 6!} + \frac{122x^8}{81 \times 8!} - \frac{4087x^{10}}{495 \times 10!}.$$

Table 2: Comparison of the residual error for various values of M by using Method 2 of Example 2.

t	Our Method			Method in [25]
	$M = 2$	$M = 3$	$M = 4$	$k = 1, M = 4$
0.1	4.33×10^{-4}	7.57×10^{-6}	5.84×10^{-6}	5.0×10^{-5}
0.2	3.42×10^{-4}	7.21×10^{-5}	4.49×10^{-6}	8.5×10^{-5}
0.3	5.81×10^{-4}	2.84×10^{-5}	3.79×10^{-6}	2.3×10^{-5}
0.4	4.10×10^{-4}	2.91×10^{-5}	5.43×10^{-6}	4.5×10^{-5}
0.5	0	5.12×10^{-5}	0	6.9×10^{-5}
0.6	4.41×10^{-4}	2.66×10^{-5}	5.60×10^{-6}	4.1×10^{-5}
0.7	6.73×10^{-4}	2.37×10^{-5}	4.04×10^{-6}	1.9×10^{-5}
0.8	4.27×10^{-4}	5.50×10^{-5}	4.95×10^{-6}	6.5×10^{-5}
0.9	5.84×10^{-4}	5.27×10^{-6}	6.65×10^{-6}	3.5×10^{-5}

The graph of compression with an exact and approximate solution and absolute error for $M = 2$ by utilizing Method 1 displayed in Figure 3. Also, In Table 2, we report the residual error for various values of M by using Method 2.

Example 3 *Now, for the last example, we consider another type of the Lane–Emden equation as follows:*

$$u^{''}(x) + \frac{2}{x}u^{'}(x) + \sinh u(x) = 0, \tag{20}$$

where, $u(0) = 1,\ u^{'}(0) = 0.$

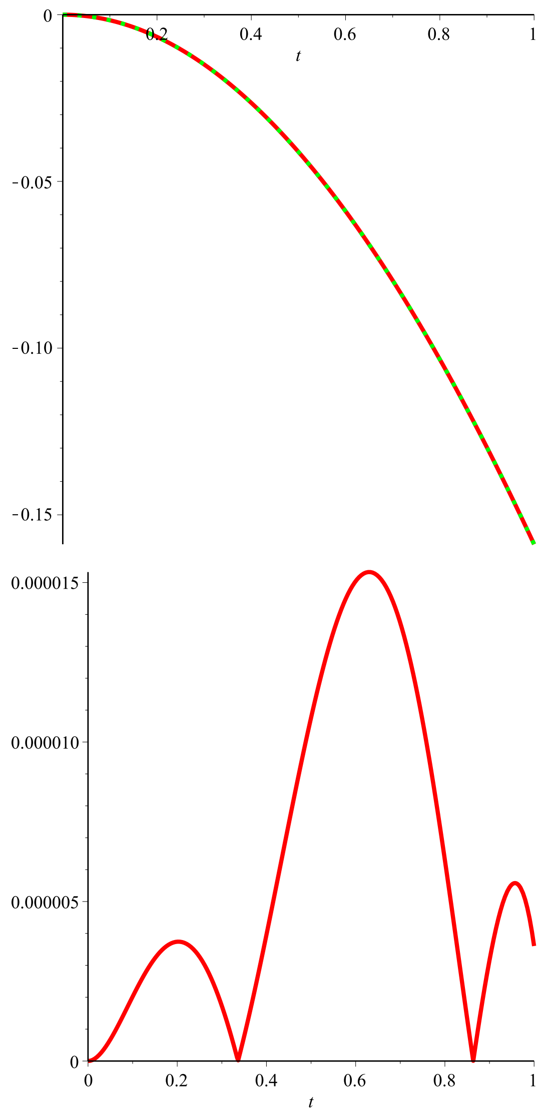

Figure 3: The compression with exact and approximate solution (above) and absolute errors (below) for approach 1 for $M = 2$ for Example 2.

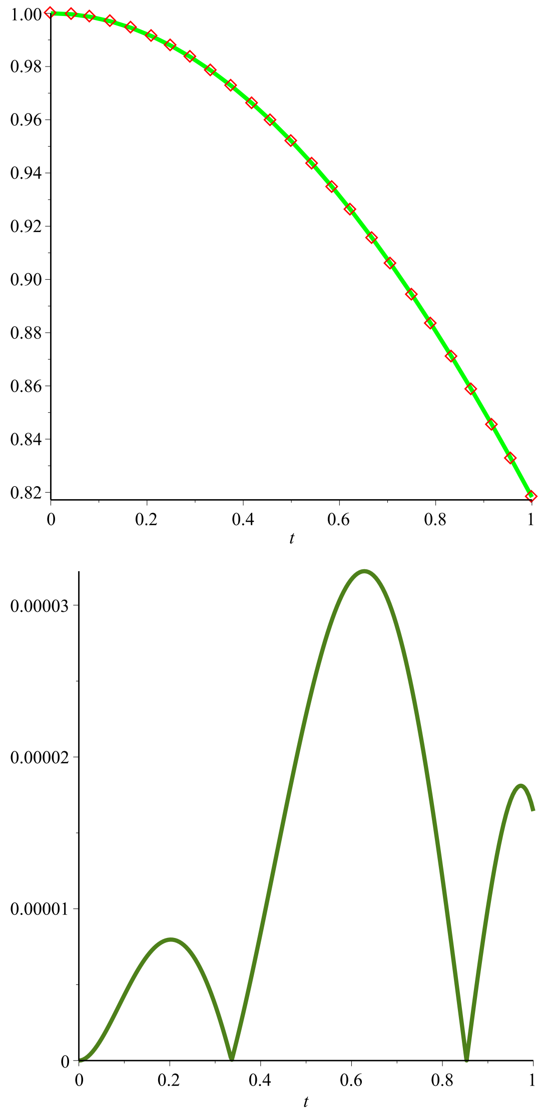

Figure 4: The compression with exact and approximate solution (above) and absolute errors (below) for approach 1 for $M = 2$ for Example 3.

We consider the solution of this problem similar to [26] as

$$u(x) \simeq 1 - \frac{e^2-1}{12e}x^2 + \frac{e^4-1}{480e^2}x^4 - \frac{2e^6+3e^2-3e^4-2}{30240e^3}x^6 + \frac{61e^8-104e^6+104e^2-61}{26127360e^4}x^8.$$

The graph of compression between exact and approximate solution (above) and absolute error (below) for $M = 2$ by utilizing Approach 1 is displayed in Figure 4. Moreover, we can see the absolute error for various numbers of M and using Method 2 in Table 3.

Table 3: Maximum absolute error by using Method 2 with different values of M of Example 3.

t	$M=2$	$M=3$	$M=4$
0.1	3.53×10^{-6}	3.62×10^{-7}	2.68×10^{-8}
0.2	6.53×10^{-6}	2.16×10^{-7}	3.45×10^{-8}
0.3	3.00×10^{-6}	6.02×10^{-7}	1.10×10^{-7}
0.4	6.89×10^{-6}	1.33×10^{-6}	8.41×10^{-8}
0.5	1.86×10^{-5}	1.37×10^{-6}	1.57×10^{-8}
0.6	2.59×10^{-5}	7.37×10^{-7}	3.61×10^{-8}
0.7	2.34×10^{-5}	9.98×10^{-8}	2.22×10^{-7}
0.8	9.87×10^{-6}	5.23×10^{-7}	1.04×10^{-6}
0.9	8.44×10^{-6}	1.43×10^{-6}	3.16×10^{-6}

6 Discussion

In this paper, we proposed two numerical techniques that utilize Chebyshev polynomials of the second kind in a neural network to obtain approximate solutions for various forms of the Lane-Emden equation. This network is made up of three layers: the input layer, the hidden layer, and the output layer. For the activation function, we consider $\mathcal{AF} = \dfrac{R}{1+|R|}, \mathcal{AF} = R$ and Chebyshev polynomials of the second kind. Ultimately, the collocation method is used to train this neural network in the first method, while the classical optimization method is applied in the second method. Numerical results show that these two approaches have the potential to become efficient algorithms for determining numerical solutions of Lane–Emden equations.

References

[1] R. P. Agarwal and D. O'Regan, Singular boundary value problems for superlinear second order ordinary and delay differential equations, *journal of differential equations* 130 (2) (1996) 333–355.

[2] R. P. Agarwal, D. O'Regan and V. Lakshmikantham, Quadratic forms and nonlinear non-resonant singular second order boundary value problems of limit circle type, *Zeitschrift für Analysis und ihre Anwendungen* 20 (3) (2001) 727–737.

[3] R. P. Agarwal and D. O'Regan, Existence theory for single and multiple solutions to singular positone boundary value problems, *Journal of Differential Equations* 175 (2) (2001) 393–414.

[4] R.P. Agarwal and D. O'Regan, Existence theory for singular initial and boundary value problems: A fixed point approach, *Applicable Analysis* 81 (2) (2002) 391–434.

[5] M. M. Coclite and G. Palmieri, On a singular nonlinear Dirichlet problem, *Communications in Partial Differential Equations* 14 (10) (1989) 1315–1327.

[6] N. T. Shawagfeh, Nonperturbative approximate solution for Lane–Emden equation, *Journal of Mathematical Physics* 34 (9) (1993) 4364-4369-

[7] H. R. Marzban, H. R. Tabrizidooz and M.Razzaghi, Hybrid functions for nonlinear initial-value problems with applications to Lane–Emden type equations, *Physics Letters A* 372 (37) (2008) 5883–5886.

[8] K. Parand, A. R. Rezaei and A.Taghavi, Lagrangian method for solving Lane–Emden type equation arising in astrophysics on semi-infinite domains, *Acta Astronautica* 67 (2010) 673–680.

[9] M. R. Eslahchi, M. Dehghan and S. Ahmadi-Asl, The general Jacobi matrix method for solving some nonlinear ordinary differential equations, *Applied Mathematical Modelling* 36 (2012) 3387–3398.

[10] S. Liao, A new analytic algorithm of Lane-Emden type equations, *Applied Mathematics and Computation* 142 (1) 1–16.

[11] R. K. Pandey, N. Kumar, Solution of Lane-Emden type equations using Bernstein operational matrix of differentiation, *New Astronomy* 17 (3) (2012) 303–308.

[12] S. A. Yousefi, Legendre wavelets method for solving differential equations of Lane-Emden type, *Applied Mathematics and Computation* 181 (2) 1417–1422.

[13] M. Lakestani and M. Dehghan, Four techniques based on theB-spline expansion and the collocation approach for the numerical solution of the Lane-Emden equation, *Mathematical Methods in the Applied Sciences* 36 (16) 2243–2253.

[14] Y. Le Cun, Y., Bengio and G. Hinton, Deep learning, *Nature* 521 (2015) 436–444.

[15] A. Malek and R. Shekari Beidokhti, Numerical solution for high order deferential equations, using a hybrid neural network-Optimization method, *Applied Mathematics and Computation* 183 (2006) 260–271.

[16] L. Jianyu, L. Siwei, Q. Yingjian and H. Yaping, Numerical solution of elliptic partial differential equation using radial basis function neural networks, *Neural Networks* 16 (2003) 729–734.

[17] Z. Masood, K. Majeed and M. Asif Zahoor Raja, Design of Mexican HatWavelet neural networks for solving Bratu type nonlinear systems, *Neurocomputing* 221 (2017) 1–14.

[18] S. Mall and S. Chakraverty, Application of legendre neural network for solving ordinary differential equations, *Applied Soft Computing* 43 (2016) 347–356.

[19] S. Mall and S. Chakraverty, Chebyshev neural network based model for solving Lane-Emden type equations, *Applied Mathematics and Computation* 247 (2014) 100–114.

[20] I. Ahmad, M. Asif Zahoor, M. Raja and F. Ashraf, Neural network methods to solve the Lane-Emden type equations arising in thermodynamic studies of the spherical gas cloud model, *Neural Computing and Applications* 28 (2017) 929–944.

[21] A. H. Hadian-Rasanan, D. Rahmati, S. Gorgin and K. Parand, A single layer fractional orthogonal neural network for solving various types of Lane-Emden equation, *New Astronomy* 75 (2020) p. 101307.

[22] A. H. Bhrawy and A. S. Alofi, The operational matrix of fractional integration for shifted Chebyshev polynomials, *Applied Mathematics Letters* 26 (2013) 25–31.

[23] E. H. Doha, A. H. Bhrawy and S. S. Ezz-Eldien , A Chebyshev spectral method based on operational matrix for initial and boundary value problems of fractional order, *Computers & Mathematics with Applications* 62 (5) (2011) 2364–2373.

[24] M. Ahmadi Darani and M. Nasiri, A fractional type of the Chebyshev polynomials for approximation of solution of linear fractional differential equations, Computational Methods for Differential Equations, 1 (2) (2013) 96–107.

[25] P. Rahimkhani and Y. Ordokhani, Orthonormal Bernoulli wavelets neural networkmethod and its application in astrophysics, *Computational and Applied Mathematics* 40 (3) (2021) p. 78.

[26] A. M. Wazwaz, A new algorithm for solving differential equations of Lane-Emden type, *Applied mathematics and computation*118 (2001) 287–310.