Math. Commun. **30** (2025), 243–258
DOI: 10.64785/mc.30.2.8

Mathematical Communications

Single-layer Laguerre neural network model for solving Lane-Emden-Fowler type equations

Sena Nur Kurmanç* and Muttalip Özavşar

Department of Mathematics, Yıldız Technical University, 34 220, Istanbul, Türkiye

Abstract. In this study, a single-layer functional link artificial neural network (FLANN) model based on Laguerre polynomials, referred to as the Laguerre Neural Network (LgNN), is utilized to solve second-order linear and non-linear equations of Lane-Emden-Fowler type. Using this model, in which the hidden layers are replaced with Laguerre polynomials, we initially expand the input patterns corresponding to a given set of nonlinear Lane-Emden-Fowler type equations. Subsequently, the network parameters are adjusted through an unsupervised error backpropagation algorithm that employs Adam optimization. Consequently, we compare the LgNN results with those obtained by other FLANN-based models, namely the Chebyshev Neural Network (ChNN) and the Legendre Neural Network (LeNN), by solving some initial value problems of Lane-Emden–Fowler type equations.

AMS subject classifications: 34-11
Keywords: Lane-Emden–Fowler equation; single-layer functional link artificial neural network; Laguerre neural network; Chebyshev neural network; Legendre neural network; Adam Optimization
Received July 6, 2023; accepted February 27, 2025

1. Introduction

Several phenomena in applied and theoretical physics can be expressed by singular nonlinear initial value problems of Lane-Emden-Fowler type equations as follows:

$$\begin{aligned} &\frac{d^2y}{dx^2} + \frac{k}{x}\frac{dy}{dx} + p(x)q(y) = r(x) \qquad x \geq 0, \\ &y(0) = a, y'(0) = b, \end{aligned} \tag{1}$$

where p, q, and r are some functions of x and y, and a, b, and k are real constants.

Furthermore, as a special class of Lane-Emden-Fowler type equations, the Lane-Emden equations apply to the thermal behavior of a spherical cloud of gas, the theory of stellar structure, the thermionic currents, and isothermal gas spheres [10, 6]. For $k = 2$, $p(x) = 1$ and $r(x) = 0$, Equation (1) takes the form of the Lane–Emden equations given below:

$$\begin{aligned} &\frac{d^2y}{dx^2} + \frac{2}{x}\frac{dy}{dx} + q(y) = 0 \qquad x \geq 0, \\ &y(0) = a, y'(0) = b. \end{aligned}$$

The existence of a singularity at the origin causes a challenge to the solution of Lane-Emden-Fowler equations and other initial value and boundary value problems in quantum mechanics and astrophysics. Hence, various techniques have been employed to solve the Lane-Emden-Fowler equations. Koch et al. [14] proposed the implicit Euler method to solve singular initial value problems. Wazwaz utilized the Adomian decomposition method and the modified decomposition method to solve equations of Emden–Fowler and Lane–Emden types [28, 29, 30, 31]. Chowdhury et al. [8, 9] solved singular initial value problems in second-order ordinary differential equations using the homotopy perturbation method. Singh et al. developed an

Email addresses: senakurmanc@gmail.com (S. N. Kurmanç), mozavsar@yildiz.edu.tr (M. Özavşar)
*Corresponding author.

effective analytic algorithm based on a modified homotopy analysis method [23]. Demir and Sungu [11] solved nonlinear singular initial value problems of Emden-Fowler type using the Differential Transformation Method (DTM) with Maple 11. To solve singular Emden-Fowler type equations, Bataineh et al. [1] proposed a homotopy analysis method. Bhrawy and Alofi [3] presented the Jacobi–Gauss collocation spectral method to solve nonlinear Lane–Emden type equations. The Haar wavelet resolution technique was introduced to solve Emden–Fowler type boundary value problems by [24]. For solving singular boundary value problems of second order, Liu et al. [16] developed a novel iterative algorithm.

As a fast-learning single-layer ANN, a functional link artificial neural network (FLANN) was described by Pao and Philips [21]. The hidden layer of FLANN is replaced with a functional expansion block that uses functions such as orthogonal polynomials, trigonometric polynomials, exponential polynomials, etc., to enhance the input patterns. The FLANN has fewer parameters than the multi-layer ANN; hence, it is more efficient in terms of computing. The FLANN has been used by several authors for solving different types of differential equations. To solve the fractional differential equation, Qu et al. employed a cosine basis function neural network [22]. Mall and Chakraverty [20] used the Legendre neural network (LeNN) to solve ordinary differential equations. To solve various types of Lane-Emden equations, Hadian-Rasanan et al. [12] utilized fractional Legendre neural networks, and Verma and Kumar [25] used MLP neural networks. With an extreme learning machine (ELM) algorithm, Hou et al. [32] presented a Legendre neural network for handling ODEs and systems of ODEs. Mall and Chakraverty [17, 18, 19] developed the Chebyshev neural network (ChNN) model to solve differential equations. Verma et. al. [27] proposed an algorithm based on ChNN and LeNN to find the solution of fractional Lienard and Duffing equations. Chakraverty [4] suggested the Laguerre neural network (LgNN) to solve a Lane-Emden equation and the Duffing oscillator equation. Chen et al. [7] presented a Laguerre neural network model for solving the Black-Scholes differential equation. Furthermore, a method based on the Hermite neural network is designed to solve Van der Pol Duffing oscillator equations in [5]. To solve the Bagley–Torvik equation, Verma and Kumar [26] used the Legendre artificial neural network.

This paper focuses on utilizing the Laguerre Neural Network (LgNN) to solve second-order linear and nonlinear equations of the Lane-Emden-Fowler type and compares the performance of LgNN with other FLANN-based models. The LgNN model is selected for its computational efficiency and accuracy, achieved through its single-layer architecture and the use of Laguerre polynomials. Here, an unsupervised version of error backpropagation with the Adam optimization technique is employed to minimize the error function and update the network parameters. The initial weights from the input to the output layer are taken as random. Furthermore, the performance of the proposed LgNN model is compared with other FLANN-based models such as the Chebyshev Neural Network (ChNN) and the Legendre Neural Network (LeNN) in terms of accuracy and error minimization. The organization of this paper is as follows. In the next section, the structure of the single-layer FLANN model and its training methodology are discussed. In section 3, we describe the ANN formulation for differential equations and, specifically, the FLANN formulation for second-order ordinary differential equations. Section 4 includes numerical examples and a comparison of existing and LgNN, ChNN, and LeNN results, and the final section is the conclusion.

2. Single-layer functional link artificial neural network

Here, we first describe the general structure of the single-layer FLANN, which provides a main frame to define distinct types of neural network models such as ChNN, LeNN, and LgNN, etc., by using special type polynomials. Then, we explain the learning algorithm of single-layer FLANN.

2.1. Structure of single-layer FLANN models

The single-layer FLANN model has a single input node, a functional expansion block based on orthogonal polynomials viz. Chebyshev, Legendre, Laguerre, Hermite, etc., and a single output node, as shown in Figure 1. In a single-layer FLANN, orthogonal polynomials $\left(U_j(x)\right)$ are used to expand the dimension of the input data. The input data is denoted by $x = (x_1, x_2, ..., x_h)^T$, where x is a single input node with h data (the number of coordinates) and p is the weight vector $p = [w_0, w_1, \ldots, w_m]$. The output of a single-layer FLANN can be written in the form:

$$N(x,p) = \phi\Big(\sum_{j=0}^{m} w_j U_j(x)\Big).$$

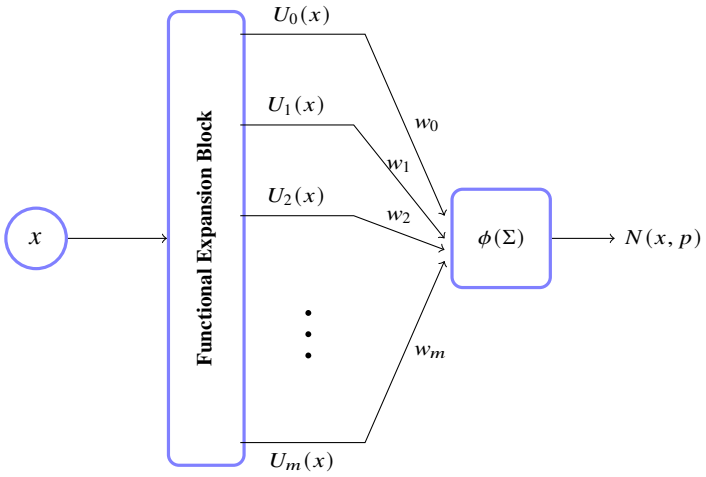


Figure 1: Structure of single-layer functional link artificial neural network

2.1.1. Laguerre neural network (LgNN)

In a Laguerre neural network, the hidden layer is removed by using Laguerre polynomials. These polynomials are derived from the Laguerre differential equation given by:

$$xy'' + (1-x)y' + \lambda y = 0.$$

We use the following recursive formula to define all Laguerre polynomials:

$$\mathbb{L}g_{n+1}(x) = (2n+1-x)\mathbb{L}g_n(x) - n^2\mathbb{L}g_{n-1}(x),$$

where $\mathbb{L}g_n(x)$ is the nth general term of the sequence $\{\mathbb{L}g_n(x)\}$ and $\mathbb{L}g_0(x) = 1, \mathbb{L}g_1(x) = -x+1$. By using Laguerre polynomials, input data $x = (x_1, x_2, ..., x_h)^T$ is transformed into a higher-dimensional space:

$$\Big[\mathbb{L}g_0(x_1), \mathbb{L}g_1(x_1), \ldots, \mathbb{L}g_m(x_1); \mathbb{L}g_0(x_2), \mathbb{L}g_1(x_2), \ldots, \mathbb{L}g_m(x_2); \ldots;$$
$$\mathbb{L}g_0(x_h), \mathbb{L}g_1(x_h), \ldots, \mathbb{L}g_m(x_h)\Big].$$

Then, Laguerre neural network's output with input x and weight vector $p = [w_0, w_1, \ldots, w_m]$ is written as stated below:

$$N(x,p) = \tanh\Big(\sum_{j=0}^{m} w_j \mathbb{L}g_j(x)\Big).$$

As an activation function, the tangent hyperbolic (tanh) function is considered in the output node of the LgNN model.

Legendre neural networks and Chebyshev neural networks have the same LgNN structure; the difference is that the functional expansion block in the hidden layer is based on either Legendre or Chebyshev polynomials. Here, $\{C_0, C_1, C_2, \ldots, C_m\}$ represent the first m Chebyshev polynomials. We can obtain the

sequence of Chebyshev polynomials thanks to the recursive formula:

$$\begin{aligned} C_0(x) &= 1, \\ C_1(x) &= x, \\ C_{k+1}(x) &= 2xC_k(x) - C_{k-1}(x). \end{aligned}$$

Also, the first few Legendre polynomials and the recursive formula for generating higher-order Legendre polynomials are given below:

$$\begin{aligned} \mathbb{L}_0(x) &= 1, \\ \mathbb{L}_1(x) &= x, \\ \mathbb{L}_2(x) &= \frac{1}{2}(3x^2 - 1), \\ \mathbb{L}_{k+1}(x) &= \frac{1}{k+1}\Big[(k+1)x\mathbb{L}_k(x) - k\mathbb{L}_{k-1}(x)\Big]. \end{aligned}$$

2.2. Learning algorithm of FLANN

In order to modify the network's parameters (weights) mentioned above and minimize the error function of the network, an unsupervised error backpropagation algorithm with Adam optimization is employed. Also, the initial weights of the single-layer FLANN model are chosen at random. As previously stated, the network output is calculated using the following formula:

$$N(x,p) = \tanh\Big(\sum_{j=0}^{m} w_j U_j(x)\Big),$$

where $x = (x_1, x_2, \ldots, x_h)^T$ is the input, U_j and w_j for $j = \{0, 1, \ldots, m\}$ are the expanded input data and the weight vector of the single-layer FLANN.

The adaptive moment estimation (Adam) algorithm is utilized to update the weights of the neural network. Adam was developed [13] as a method for efficient stochastic optimization that requires only first-order gradients with little memory demand. To update the weights, Adam computes the first moment m_t and second moment v_t of the gradient:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) {g_t}^2, \end{aligned} \tag{2}$$

where $\beta_1, \beta_2 \in [0, 1)$ are the exponential decay rates for the moment estimates, g_t indicates the vector of partial derivatives of error function $E(x, p)$, with respect to w_j evaluated at timestep t and g_t^2 denotes the $g_t \odot g_t$ (elementwise matrix multiplication $\odot$ between g_t and g_t).

Since m_t and v_t are initialized as 0 vectors, the moment estimates are biased towards zero. These biases are counteracted by computing bias-corrected estimates $\hat{m_t}$ and $\hat{v_t}$ as follows:

$$\begin{aligned} \hat{m_t} &= \frac{m_t}{1-\beta_1^t} \\ \hat{v_t} &= \frac{v_t}{1-\beta_2^t}, \end{aligned} \tag{3}$$

where β_1^t and β_2^t stand for β_1 and β_2 to the power of t. Finally, the weights of FLANN are updated as follows:

$$w_j^{t+1} = w_j^t - \frac{\eta}{\sqrt{\hat{v_t}} + \epsilon}\hat{m_t}, \tag{4}$$

where t is the iteration step and η is the value of stepsize. Kingma and Ba [13] suggest that to improve the efficiency of the method in machine learning, one should implement the following values: $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.

3. A neural network method to solve differential equations

This section examines the general formulation of differential equations using the ANN, with a focus on the FLANN formulation for second-order ordinary differential equations.

A differential equation has the following general form [15]:

$$G\left(\vec{x}, y(\vec{x}), \nabla y(\vec{x}), \nabla^2 y(\vec{x}), \ldots, \nabla^k y(\vec{x})\right) = 0, \qquad \vec{x} \in D \subset \mathbb{R}^n,$$

subject to initial/boundary conditions, where $\vec{x} = (x_1, x_2, ..., x_n) \in \mathbb{R}^n$, G is the function that defines the differential equation's structure, $y(\vec{x})$ stands for the solution and ∇ is a differential operator.
To solve the differential equation mentioned above, the domain D and its boundary S discretize into a collection of discrete points $\hat{D}$ and $\hat{S}$. Then, the following equation system is used to solve the problem:

$$G(\vec{x}_i, y(\vec{x}_i), \nabla y(\vec{x}_i), \nabla^2 y(\vec{x}_i), \ldots, \nabla^n y(\vec{x}_i)) = 0, \quad \forall \vec{x}_i \in \hat{D},$$

along with the boundary conditions. Let $y_t(\vec{x}, \vec{p})$ represent the trial solution with the adjustable parameter $\vec{p}$; the problem is converted into the following minimization problem:

$$\min_{\vec{p}} \sum_{\vec{x}_i \in \hat{D}} \frac{1}{2}\left(G\left(\vec{x}_i, y_t(\vec{x}_i, \vec{p}), \nabla y_t(\vec{x}_i, \vec{p}), \nabla^2 y_t(\vec{x}_i, \vec{p}), \ldots, \nabla^k y_t(\vec{x}_i, \vec{p})\right)\right)^2.$$

The trial solution with parameters $\vec{p}$ and input vector $\vec{x}$ can be written as

$$y_t(\vec{x}, \vec{p}) = A(\vec{x}) + F(\vec{x}, N(\vec{x}, \vec{p})),$$

where $A(\vec{x})$ meets initial or boundary conditions and does not have any adjustable parameters, and $F(\vec{x}, N(\vec{x}, \vec{p}))$ contains $N(\vec{x}, \vec{p})$ single output of the feedforward neural network.

3.1. FLANN formulation for second-order ODE

Since our goal is to solve Lane-Emden-Fowler type equations, we will closely examine the formulation of the second-order ODEs.

Let us consider a general second-order ordinary differential equation

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right) \qquad x \in [a, b], \tag{5}$$

with $y(a) = B, y'(a) = B'$.
The trial solution for equation (5) can be formulated as

$$y_t(x, p) = B + B'(x - a) + (x - a)^2 N(x, p),$$

where $N(x, p)$ represents the single-layer FLANN output with input data x and adjustable network parameters p.
As previously stated, the FLANN output $N(x, p)$ is

$$N(x, p) = \tanh(z),$$

where $z = \sum_{j=0}^{m} w_j U_j(x)$ and w_j are weights of the FLANN, and $U_j(x)$ indicates the corresponding orthogonal polynomials.
Then, the error function for the second-order ODE that has to be minimized can be written as follows:

$$E(x, p) = \sum_{i=1}^{h} \frac{1}{2}\left\{\frac{d^2 y_t(x_i, p)}{dx^2} - f\left(x_i, y_t(x_i, p), \frac{dy_t(x_i, p)}{dx}\right)\right\}^2.$$

To minimize the error function, we employed the Adam algorithm, as shown in equations (2)-(4).

4. Numerical results

In this section, the Laguerre neural network (LgNN), the Chebyshev neural network (ChNN), and the Legendre neural network (LeNN) are employed to solve four initial value problems of Lane-Emden-Fowler type equations. The first two experiments are the standard Lane-Emden equations. All results are computed using Python 3.8.
For $q(y) = y^m$, the Lane-Emden equation transforms to the standard Lane–Emden equation, which has the form:

$$\frac{d^2y}{dx^2} + \frac{2}{x}\frac{dy}{dx} + y^m = 0, \qquad x \geq 0. \tag{6}$$

Equations of the form 6 provide an efficient model for the thermal behavior of a spherical cloud of gas acting in accordance with the classical laws of thermodynamics and under the mutual attraction of its molecules [10]. It has been demonstrated in [6] that an analytic solution of (6) exists just for $m = 0, 1$, and 5.

Example 1. First, we consider the standard Lane–Emden equation for $m = 1$

$$y'' + \frac{2}{x}y' + y = 0, \tag{7}$$

with $y(0) = 1$, $y'(0) = 0$.

The exact solution to (7) is as follows [10, 2]:

$$y(x) = \frac{\sin(x)}{x}.$$

The FLANN trial solution of (7) can be stated as follows:

$$y_t(x, p) = 1 + x^2 N(x, p).$$

FLANN models are trained with twenty equidistant points in the range $[0, 1]$ and five weights. The comparison between the analytical results and the results of LgNN, ChNN, and LeNN is shown in Figure 2. Figure 3 displays the error between analytical results and LgNN, ChNN, and LeNN results, respectively. Also, mean absolute errors between analytical results and FLANN model results are given in Table 1.

	LgNN	**ChNN**	**LeNN**
MAE	3.678450e-05	4.549000e-04	3.437355e-04

Table 1: Mean absolute errors between analytical results and FLANN results (Example 1)

Example 2. Now, we take the standard Lane–Emden equation with $m = 5$

$$y'' + \frac{2}{x}y' + y^5 = 0 \qquad x \geq 0, \tag{8}$$

subject to $y(0) = 1$, $y'(0) = 0$.

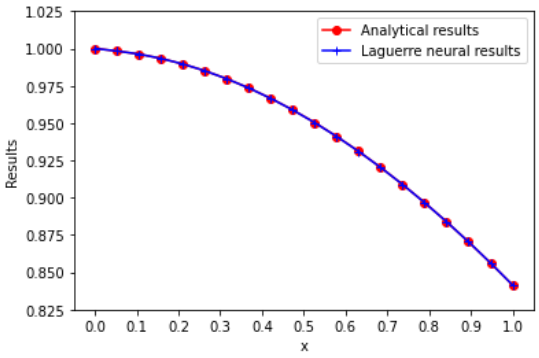
Equation 8 has a particular solution [10] for $x \geq 0$, which is as follows:

$$y(x) = \left(1 + \frac{x^2}{3}\right)^{-1/2}.$$

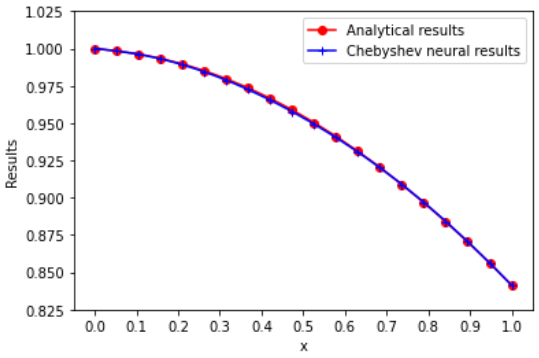
We can write the FLANN trail solution of (8) as

$$y_t(x, p) = 1 + x^2 N(x, p).$$

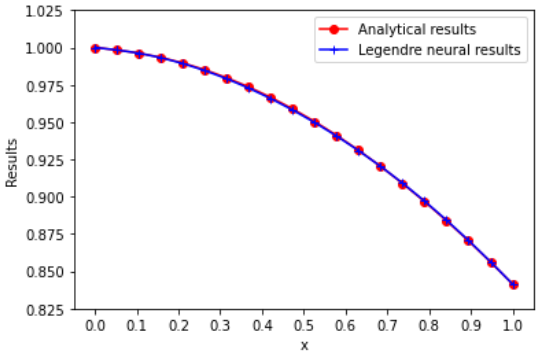
For computing the results, we trained the functional link artificial networks for 10 evenly spaced points in the range $[0, 1]$ with 5 weights from the input layer to the output layer. Analytical, LgNN, ChNN, and LeNN results and absolute errors between them are shown in Figures 4 and 5. Mean absolute errors between analytical results and FLANN model results are given in Table 2.



(a) LgNN



(b) ChNN



(c) LeNN

Figure 2: Comparisons of analytical results with the LgNN, ChNN, and LeNN results, respectively (Example 1)

	LgNN	ChNN	LeNN
MAE	7.779200e-05	1.351930e-04	5.752020e-04

Table 2: Mean absolute errors between analytical results and FLANN results (Example 2)

Example 3. In this example, let us take a nonlinear Lane-Emden–Fowler equation

$$y'' + \frac{3}{x}y' + 2x^2y^2 = 0,$$

subject to $y(0) = 1, y'(0) = 0$.

In this instance, the FLANN solution can be formulated as

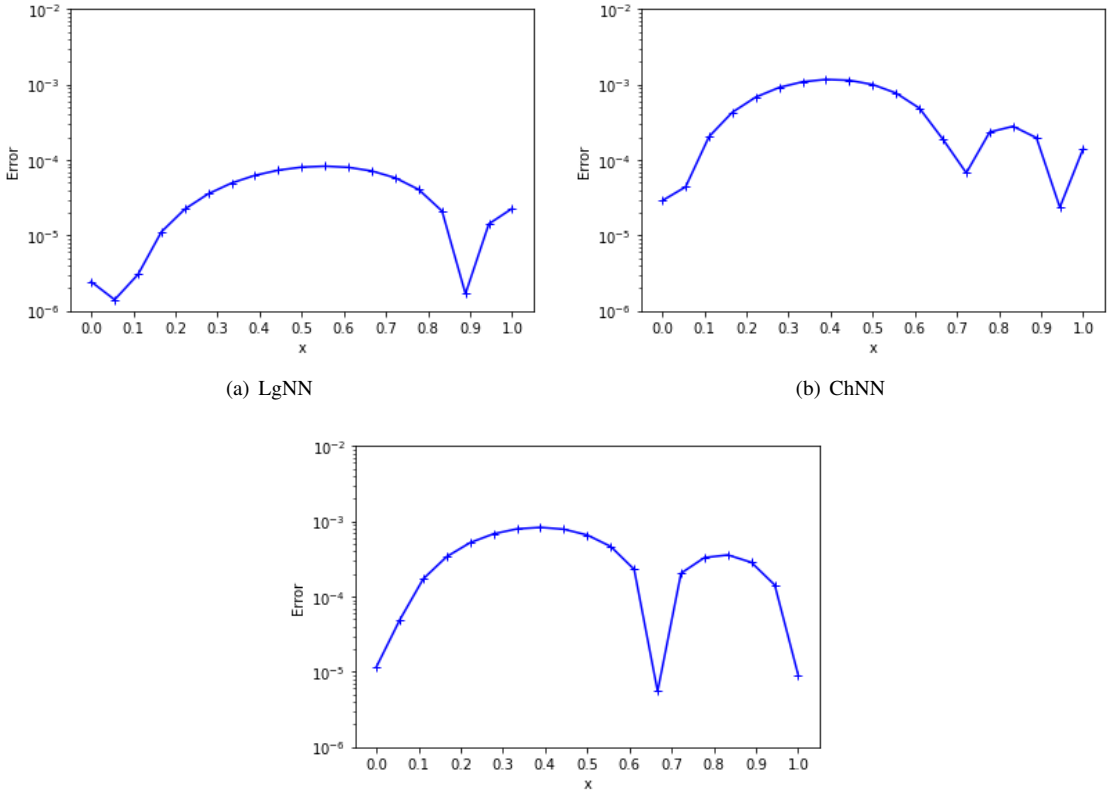
$$y_t(x, p) = 1 + x^2N(x, p).$$

Ten equidistant points in $[0, 1]$ are used to train the network. The Laguerre neural network with the first six Laguerre polynomials, as well as ChNN and LeNN with the first five Chebyshev and Legendre polynomials, are considered. Figure 6 includes the comparison among solutions obtained by Maple 11 [11], LgNN, ChNN, and LeNN. The plot of the error between Maple 11 [11] and LgNN, ChNN, and LeNN solutions, respectively, is given in Figure 7.

Example 4. Finally, we analyze the nonlinear, nonhomogeneous Lane-Emden–Fowler equation.

$$y'' + \frac{8}{x}y' + xy^2 = x^4 + x^5, \qquad x \geq 0 \tag{9}$$

with $y(0) = 1, y'(0) = 0$.



(a) LgNN

(b) ChNN

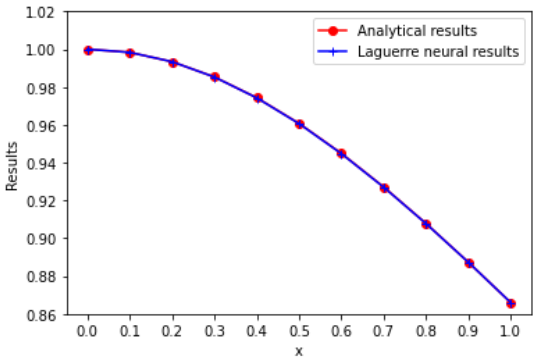
(c) LeNN

Figure 3: Plots of errors between analytical results and LgNN, ChNN, and LeNN results, respectively (Example 1)

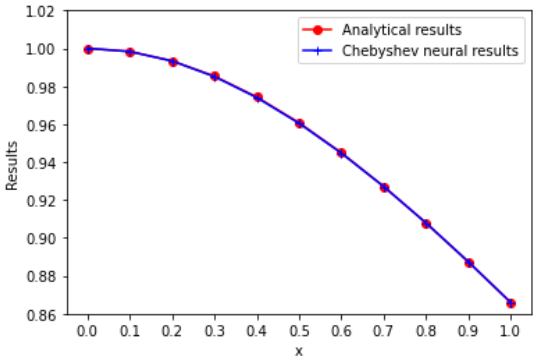
The related FLANN trial solution can be written as

$$y_t(x, p) = 1 + x^2 N(x, p).$$

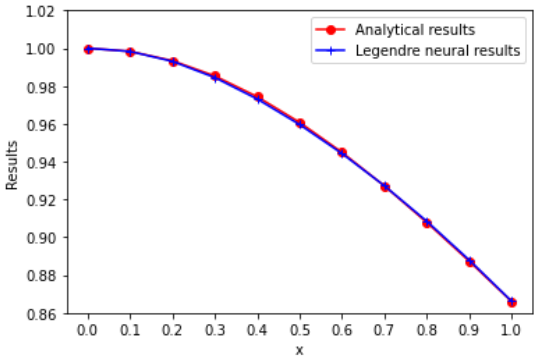
To solve Equation 9, ten points in the domain $[0, 1]$ are taken into account, and the Chebyshev neural network with the first six Chebyshev polynomials and LgNN and LeNN with the first five Laguerre and Legendre polynomials are used. A comparison among the numerical solutions obtained by Maple 11 [11], LgNN, ChNN, and LeNN is shown in Figure 8. Figure 9 illustrates the plot of error between Maple 11 [11] and, respectively, the LgNN, ChNN, and LeNN results.



(a) LgNN

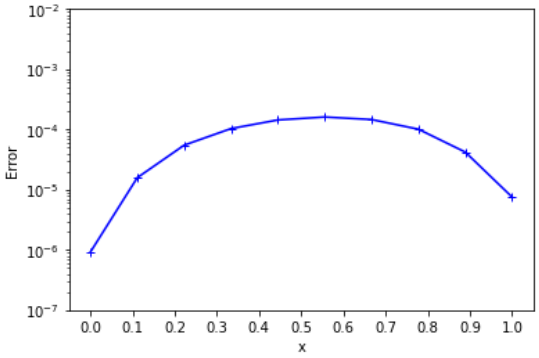


(b) ChNN

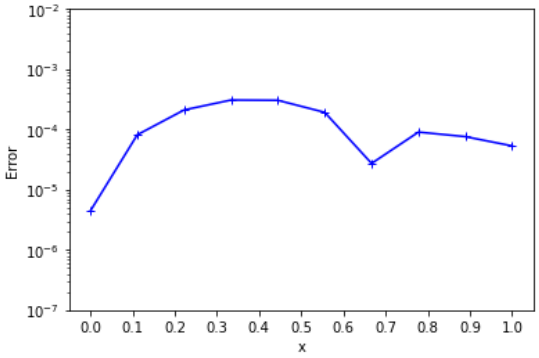


(c) LeNN

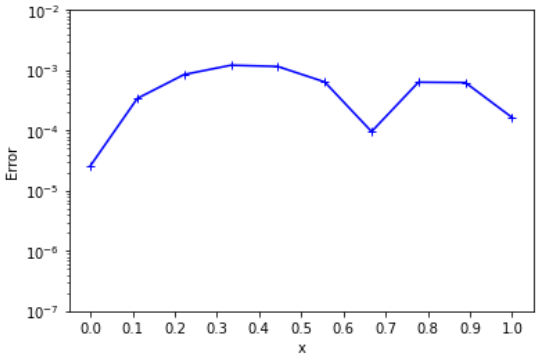
Figure 4: Comparison of analytical results with LgNN, ChNN, and LeNN results, respectively (Example 2)



(a) LgNN

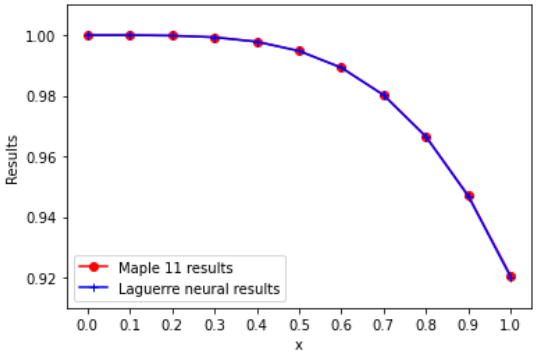


(b) ChNN

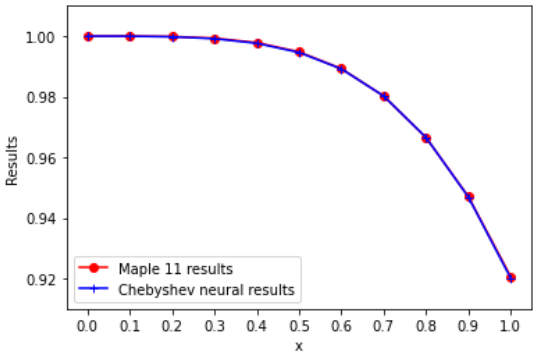


(c) LeNN

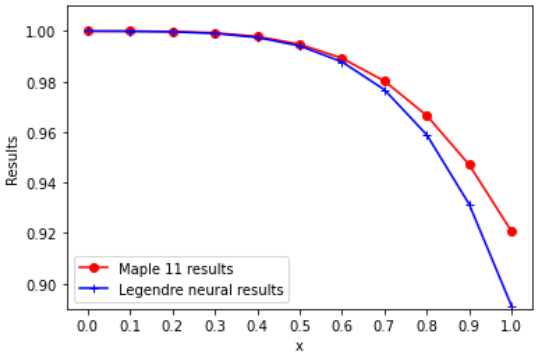
Figure 5: Plots of errors between analytical results and LgNN, ChNN, and LeNN results, respectively (Example 2)



(a) LgNN

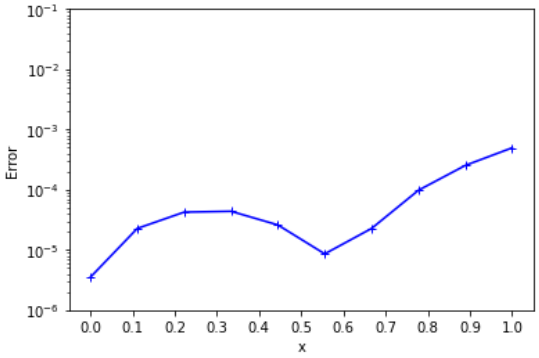


(b) ChNN

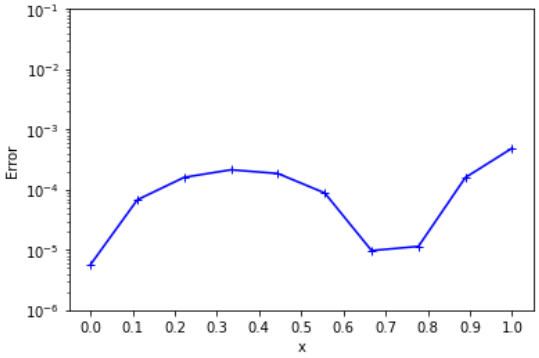


(c) LeNN

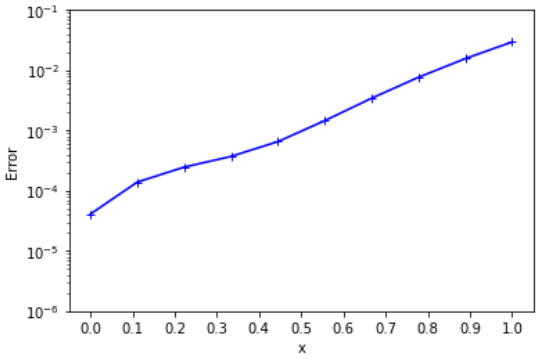
Figure 6: Comparison of Maple 11 [11] results with LgNN, ChNN, and LeNN results, respectively (Example 3)



(a) LgNN

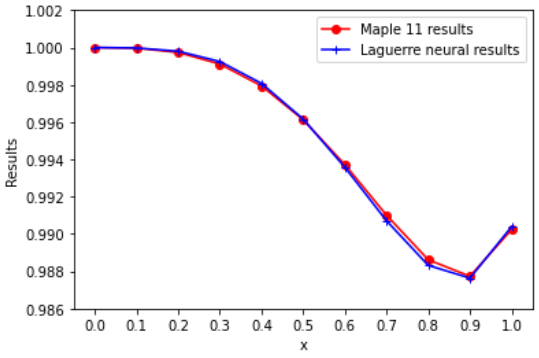


(b) ChNN

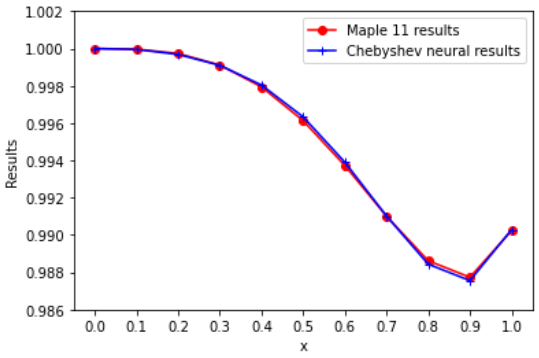


(c) LeNN

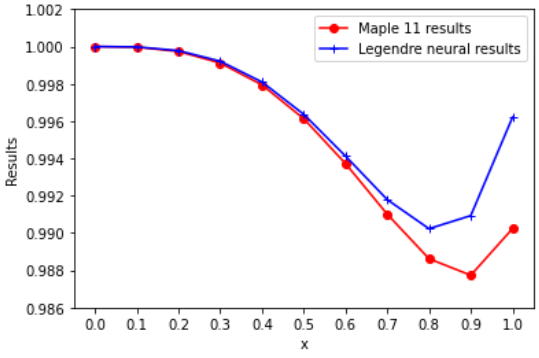
Figure 7: Plots of errors between Maple 11 [11] results and LgNN, ChNN, and LeNN results, respectively (Example 3)



(a) LgNN

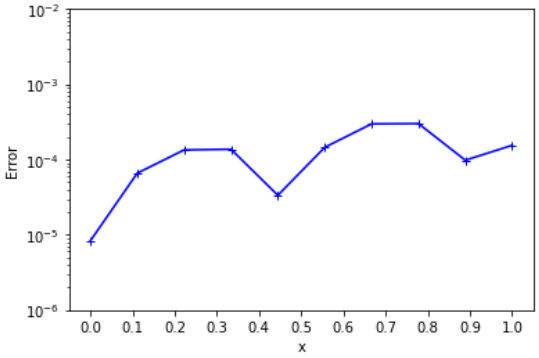


(b) ChNN

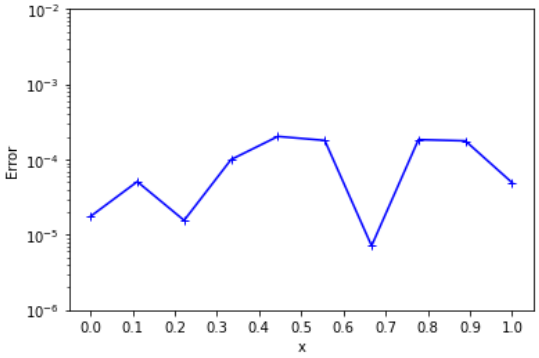


(c) LeNN

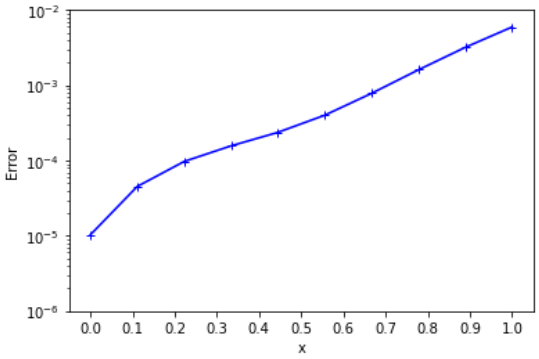
Figure 8: Comparison of Maple 11 [11] results with the results of LgNN, ChNN, and LeNN, respectively (Example 4)



(a) LgNN



(b) ChNN



(c) LeNN

Figure 9: Plots of errors between Maple 11 [11] results and LgNN, ChNN, and LeNN results, respectively (Example 4)

5. Conclusion

In this paper, a single-layer functional link artificial neural network (FLANN) model, that is, the Laguerre artificial neural network (LgNN), is applied to linear and nonlinear initial value problems of Lane-Emden-Fowler type equations. To solve the difficulty of Lane-Emden-Fowler-type equations due to the presence of a singularity at $x = 0$, the single-layer Laguerre artificial neural network (LgNN) is used here. The hidden layer in LgNN is a functional expansion block that uses Laguerre polynomials to enhance the input patterns. Its single-layer architecture reduces computational complexity, making it more efficient than multi-layer neural networks. Moreover, the LgNN model is flexible enough to handle both linear and nonlinear Lane-Emden-Fowler equations, whereas some traditional methods are limited to specific cases or require significant modifications for nonlinear problems. In conclusion, this study demonstrates the potential of the Laguerre Neural Network (LgNN) as a powerful and efficient tool for solving Lane-Emden-Fowler type equations. By addressing the challenges posed by singularities and nonlinearities, the LgNN model provides accurate and computationally efficient solutions. Additionally, the study employs the Adam optimization algorithm, which ensures faster convergence and better performance compared to traditional optimization methods. The study also conducts a systematic comparison of the performance of the LgNN model against other FLANN-based models, including the Chebyshev Neural Network (ChNN) and the Legendre Neural Network (LeNN). Figures and tables are used to display the numerical results of the LgNN, ChNN, and LeNN models. The results of the numerical studies show that the LgNN and ChNN are more accurate than the LeNN model. Especially when we compare the errors between analytical results and FLANN models, the results emphasize the advantages of the LgNN model in terms of accuracy and power. While the study primarily focuses on second-order ordinary differential equations (ODEs) of the Lane-Emden-Fowler type, the LgNN's architecture and methodology can be extended to PDEs, which are widely used in various scientific and engineering fields. Its ability to handle nonlinearities, singularities, and complex boundary conditions makes it a versatile and efficient approach for addressing the challenges posed by PDEs in theoretical and applied research.

References

[1] A. S. BATAINEH, M. S. M. NOORANI, and I. HASHIM, Homotopy analysis method for singular IVPs of Emden-Fowler type, *Commun. Nonlinear Sci. Numer. Simul.* **14** (2009), no. 4, 1121–1131.

[2] C. M. BENDER, K. A. MILTON, S. S. PINSKY, and L. M. SIMMONS, JR., A new perturbative approach to nonlinear problems, *J. Math. Phys.* **30** (1989), no. 7, 1447–1455.

[3] A. H. BHRAWY and A. S. ALOFI, A Jacobi-Gauss collocation method for solving nonlinear Lane-Emden type equations, *Commun. Nonlinear Sci. Numer. Simul.* **17** (2012), no. 1, 62–70.

[4] S. CHAKRAVERTY (ed.), *Mathematical methods in interdisciplinary sciences*, John Wiley & Sons, Inc., 2020.

[5] S. CHAKRAVERTY and S. MALL, *Artificial neural networks for engineers and scientists: solving ordinary differential equations*, CRC Press, Boca Raton, 2017.

[6] S. CHANDRASEKHAR, *An introduction to the study of stellar structure*, Dover Publications, Inc., New York, 1957.

[7] Y. CHEN, H. YU, X. MENG, X. XIE, M. HOU, and J. CHEVALLIER, Numerical solving of the generalized Black-Scholes differential equation using Laguerre neural network, *Digital Signal Processing* **112** (2021), 103003.

[8] M. CHOWDHURY and I. HASHIM, Solutions of a class of singular second-order IVPs by homotopy-perturbation method, *Phys. Lett. A* **365** (2007), no. 5-6, 439–447.

[9] M. CHOWDHURY and I. HASHIM, Solutions of Emden-Fowler equations by homotopy-perturbation method, *Nonlinear Anal. Real World Appl.* **10** (2009), no. 1, 104–115.

[10] H. T. DAVIS, *Introduction to nonlinear differential and integral equations*, US Government Printing Office, Washington, DC, 1961.

[11] H. DEMIR and I. Ç. SÜNGÜ, Numerical solution of a class of nonlinear Emden-Fowler equations by using differential transform method, *Cankaya University Journal of Science and Engineering* **12** (2009), no. 2, 75–82.

[12] 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), 101307.

[13] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, (2014), arXiv preprint: arXiv:1412.6980. Available at https://arxiv.org/abs/1412.6980.

[14] O. Koch, P. Kofler, and E. B. Weinmüller, The implicit Euler method for the numerical solution of singular initial value problems, *Appl. Numer. Math.* **34** (2000), no. 2-3, 231–252.

[15] E. Lagaris, I, A. Likas, and I. Fotiadis, D, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Transactions on Neural Networks* **9** (1998), no. 5, 987–1000.

[16] C.-S. Liu, E. R. El-Zahar, and C.-W. Chang, A boundary shape function iterative method for solving nonlinear singular boundary value problems, *Math. Comput. Simulation* **187** (2021), 614–629.

[17] S. Mall and S. Chakraverty, Chebyshev neural network based model for solving Lane-Emden type equations, *Appl. Math. Comput.* **247** (2014), 100–114.

[18] S. Mall and S. Chakraverty, Numerical solution of nonlinear singular initial value problems of Emden-Fowler type using Chebyshev neural network method, *Neurocomputing* **149** (2015), 975–982.

[19] S. Mall and S. Chakraverty, Single layer Chebyshev neural network model with regression-based weights for solving nonlinear ordinary differential equations, *Evolutionary Intelligence* **13** (2015), no. 4, 687–694.

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

[21] Y.-H. Pao and S. M. Phillips, The functional link net and learning optimal control, *Neurocomputing* **9** (1995), no. 2, 149–164, Control and Robotics, Part II.

[22] H. Qu and X. Liu, A numerical method for solving fractional differential equations by using neural network, *Applied Soft Computing* (2016), 347–3562.

[23] O. P. Singh, R. K. Pandey, and V. K. Singh, An analytic algorithm of Lane-Emden type equations arising in astrophysics using modified homotopy analysis method, *Comput. Phys. Comm.* **180** (2009), no. 7, 1116–1124.

[24] R. Singh, V. Guleria, and M. Singh, Haar wavelet quasilinearization method for numerical solution of Emden-Fowler type equations, *Math. Comput. Simulation* **174** (2020), 123–133.

[25] A. Verma and M. Kumar, Numerical solution of Lane-Emden type equations using multilayer perceptron neural network method, *Int. J. Appl. Comput. Math.* **5** (2019), no. 5, Paper no. 141.

[26] A. Verma and M. Kumar, Numerical solution of Bagley-Torvik equations using Legendre artificial neural network method, *Evolutionary Intelligence* **14** (2021), 2027–2037.

[27] A. Verma, W. Sumelka, and P. K. Yadav, The numerical solution of nonlinear fractional lienard and duffing equations using orthogonal perceptron, *Symmetry* **15** (2023), no. 9, Paper no. 1753.

[28] A.-M. Wazwaz, A new algorithm for solving differential equations of Lane-Emden type, *Appl. Math. Comput.* **118** (2001), no. 2-3, 287–310.

[29] A.-M. Wazwaz, Adomian decomposition method for a reliable treatment of the Emden-Fowler equation, *Appl. Math. Comput.* **161** (2005), no. 2, 543–560.

[30] A.-M. Wazwaz, The modified decomposition method for analytic treatment of differential equations, *Appl. Math. Comput.* **173** (2006), no. 1, 165–176.

[31] A.-M. Wazwaz, R. Rach, and J.-S. Duan, A study on the systems of the Volterra integral forms of the Lane-Emden equations by the Adomian decomposition method, *Math. Methods Appl. Sci.* **37** (2014), no. 1, 10–19.

[32] Y. Yang, M. Hou, and J. Luo, A novel improved extreme learning machine algorithm in solving ordinary differential equations by Legendre neural network methods, *Adv. Difference Equ.* (2018), 1–24.