Numerical Solution of Sixth-Order Differential Equations Arising in Astrophysics by Neural Network

M. Khalid
Department of Mathematical Sciences
Federal Urdu University of Arts, Sciences & Techonology
University Road, Karachi-75300, Pakistan

Mariam Sultana
Department of Mathematical Sciences
Federal Urdu University of Arts, Sciences & Technology
University Road, Karachi-75300, Pakistan

Faheem Zaidi
Department of Mathematical Sciences
Federal Urdu University of Arts, Sciences & Technology
University Road, Karachi-75300, Pakistan

ABSTRACT

In the current paper, a neural network method to solve sixth-order differential equations and their boundary conditions has been presented. The idea this method incorporates is to integrate knowledge about the differential equation and its boundary conditions into neural networks and the training sets. Neural networks are being used incessantly to solve all kinds of problems hailing a wide range of disciplines. Several examples are given to illustrate the efficiency and implementation of the Neural Network method.

Keywords:

Sixth Order Differential Equation, Boundary Conditions, Neural Network

1. INTRODUCTION

Neural networks of an artificial nature are an interesting kind of artificial intelligence; they are unique in itself. Neural networks mimic the learning process of the human brain in order to extract patterns from historical data [1]. For a great number of years this new type of expertise has been successfully employed to a variety of real-world problems [2]. The first introduction of perceptrons was proposed by Rosenblatt [3]. Simple perceptrons are supervised networks; in the case that is largely unsupervised, the network completely acclimatizes to its inputs and furthers itself accordingly. These networks can be taught to distinguish structures and patterns from their input. Multi-layered perceptrons (those that have more than three layers) use multiple hidden layers.

Astrophysics gives rise to many sixth-order boundary-value problems; the thin layers that convect are bounded by stable layers, and the former are known to surround A-type stars. This may be modeled by sixth-order boundary-value problems [4, 5]. Numerical analysis literature proves that very little work has been conducted on finding the solution for sixth-order boundary-value problems [4–8]. A thorough discussion has been presented regarding theorems that list conditions for the existence and uniqueness of solutions of such problems in [9].

Looking at it from the perspective of numerals, Shen [10], Doha and Bhrawy [11–13], and Doha et al. [14] have succeeded in constructing a competent Spectral-Galerkin algorithms that make use of packed combinations of orthogonal polynomials to solve elliptic equations of the second and fourth order with coefficients that remain constant in various situations. In [15–17] the respective authors have deigned to present efficient Jacobi dual-Petrov-Galerkin and Jacobi-Gauss methods of collocation to find solutions to differential equations of some odd-order. More so, the Bernstein polynomials have been employed for conducting a numerical solution of differential equations of high even-order [18, 19]. Siddiqi and Twizell [7] made use of sixth-degree Splines, where the value of Spline at the mid knots of the interpolation interval were related through consistency relations to the corresponding values of the even order derivatives. A B-Spline function of sixth-degree is employed for the processing of an approximate solution for boundary-value problems of the sixth order [20]. Furthermore, another set of sixth-order problem related solutions are introduced, called Septic Spline solutions [21]. Another milestone was sixth-order differential equations by Twizell and Boutayeb [5]; they developed finite-difference methods of order two, four, six, and eight for the solution of problems as these. Authors in [8] gave rise to the Sinc-Galerkin method to solve sixth-order boundary-value problems. Wazwaz [22] used decomposition and modified domain decomposition methods for the same purpose and Bhrawy [23] put forward a spectral Legendre-Galerkin method to solve sixth-order boundary-value problems.

In this work, sixth order differential equations with boundary conditions are mathematically modeled with feed-forward artificial neural network with some adaptive parameters. The paper has been organized as follows: Section 2 summarizes and discusses papers that address neural network in solving Differential Equation; the objective function creation with neural network mathematical modeling and its training methodology are introduced in Section 3. A detailed application of the designed method along with some discussion on the results is presented in Section 4. Comparative analysis of the results, conclusively, is presented in Section 5.

2. LITERATURE REVIEW

Differential equations play an imperative role in various fields of engineering and science. The exact solution of differential equations may not be always possible. For this reason, then, various types of well known numerical methods such as Euler, Runge-Kutta, Predictor-Corrector, finite element, and finite difference methods, are incorporated for solving these equations. Although these numerical methods provide good approximations to the solution, they may be challenging for problems on a higher dimension. In recent years, many researchers tried to devise new methods for solving differential equations. As such are Artificial Neural Network (ANN) based models are used to solve ordinary differential equations with initial conditions.

Lee & Kang [24] initially introduced a method to solve first order differential equation using Hopfield neural network models. Then, another approach by Meade & Fernandez [25, 26] was proposed for both linear and non-linear differential equations using Splines and feed forward neural network. Artificial neural networks based on Broyden-Fletcher-Goldfarb-Shanno ($BFGS$) optimization technique for solving ordinary and partial differential equations have been excellently presented by Lagaris et al. [27]. Furthermore, Lagaris et al. [28] investigated neural network methods for boundary value problems with irregular boundaries. Parisi et al. [29] presented unsupervised feed forward neural networks for the solution of differential equations. The potential of the hybrid and optimization technique to deal with differential equation of lower order as well as higher order has been presented by Malek & Shekari Beidokhti [30]. Choi & Lee [31] discussed comparison of generalizing ability on solving differential equation using back propagation and reformulated radial basis function network. Yazdi et al. [32] used unsupervised kernel least mean square algorithm for solving ordinary differential equations. A new algorithm for solving matrix Riccati differential equations has been developed and employed by Selvaraju & Abdul Samant [33]. He et al. [34] investigated a class of partial differential equations using multilayer neural network. Kumar & Yadav [35] surveyed multilayer perceptrons and radial basis function neural network methods for the solution of differential equations. Tsoulos et al. [36] solved differential equations with neural networks using a scheme that worked on the basis of grammatical evolution. Numerical solution of elliptic partial differential equation using radial basis function neural networks has been presented by Jianyu et al. [37]. Shirvany et al. [38] proposed multilayer perceptron and radial basis function (RBF) neural networks with a new unsupervised training method for numerical solution of partial differential equations. Mai-Duy & Tran-Cong [39] discussed numerical solution of differential equations using multiquadric radial basis function networks. Fuzzy linguistic model in neural network to solve differential equations is presented by Leephakpreeda [40]. Franke & Schaback [41] solved partial differential equations by collocation using radial basis functions. Smaoui & Al-Enezi [42] presented the dynamics of two non-linear partial differential equations using artificial neural networks. Differential equations with genetic programming have been analyzed by Tsoulos & Lagaris [43]. McFall & Mahan [44] used artificial neural network for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions. Hoda & Nagla [45] solved mixed boundary value problems using multilayer perceptron neural network method.

As per the review of the literatures, it reveals that authors have taken the parameters (weights/biases) as arbitrary (random) and the numbers of nodes in the hidden layer are considered by a trial and error method. In this paper, a method for solving sixth-order differential equations using feed forward neural network is proposed as a basic approximation element and error back propagation algorithm [46, 47] by fixing hidden nodes as per the required accuracy. The trial solution of the model is generated by training the algorithm. The approximate solution by ANN has numerous benefits compared with traditional numerical methods. The neural network trial solution is written as sum of two terms; the first one satisfies initial/boundary conditions and the second part involves neural network with adjustable parameters.

3. DESCRIPTION OF THE METHOD

The generic form of sixth order differential equations may be represented as

$$\frac{d^6y(x)}{dx^6} = f(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \frac{d^3y}{dx^3}, \frac{d^4y}{dx^4}, \frac{d^5y}{dx^5}) \quad x \in [a, b] \tag{1}$$

subject to boundary condition

$$\begin{aligned} y(a) &= A; \quad y^{''}(a) = C; \quad y^{iv}(a) = E \\ y(b) &= B; \quad y^{''}(b) = D; \quad y^{iv}(b) = F \end{aligned} \tag{2}$$

Assuming that the trial solution is of the form

$$y_T(x, p) = J(x) + M(x)N(x, p) \tag{3}$$

There are any forms for functions $J(x)$ and $M(x)$ to construct $y_T(x, p)$. Sigmoid function $\tanh(b_i + c_i x)$ is accepted to be an activation function for hidden unit. The trial function can not be satisfied in the boundary conditions unless there is

$$\begin{aligned} J(a) &= A; \quad J^{''}(a) = C; \quad J^{iv}(a) = E \\ J(b) &= B; \quad J^{''}(b) = D; \quad J^{iv}(b) = F \end{aligned} \tag{4}$$

The function $M(x)$ is taken as $(x-a)^3(x-b)^3$. This motivates the introduction of $J(x) = w_o + w_1x + w_2x^2 + ... + w_5x^5$ which is polynomial of five degree with coefficient $w_o, w_1, ...w_5$

$$M(a).N(a, p) = 0 \tag{5}$$

$$M(b).N(b, p) = 0 \tag{6}$$

$$M(a).N^{''}(a, p) + 2M^{'}(a).N^{'}(a, p) + M^{''}(a).N(a, p) = 0 \tag{7}$$

$$M(b).N^{''}(b, p) + 2M^{'}(b).N^{'}(b, p) + M^{''}(b).N(b, p) = 0 \tag{8}$$

$$\begin{aligned} &M^{iv}(a).N(a, p) + 4M^{'''}(a).N^{'}(a, p) + 6M^{''}(a).N^{''}(a, p) + \\ &4M^{'}(a).N^{'''}(a, p) + M(a).N^{iv}(a, p) = 0 \end{aligned} \tag{9}$$

$$\begin{aligned} &M^{iv}(b).N(b, p) + 4M^{'''}(b).N^{'}(b, p) + 6M^{''}(b).N^{''}(b, p) + \\ &4M^{'}(b).N^{'''}(b, p) + M(b).N^{iv}(b, p) = 0 \end{aligned} \tag{10}$$

The following conditions must be satisfied with Eq. 3

$$w_o + w_1a + w_2a^2 + w_3a^3 + w_4a^4 + w_5a^5 = A \tag{11}$$

Table 1. Values of Neural parameters (Example 1), a_i, b_i and c_i with five nodes

i	a_i	b_i	c_i
1	0.016505561	-0.2434469	0.863716
2	0.087265869	-0.0058665	1.069340
3	0.024116433	-0.0300997	1.0776682
4	0.135271789	-0.0075984	0.7272999
5	-0.19112553	-0.1365936	1.4643935

$$w_o + w_1 b + w_2 b^2 + w_3 b^3 + w_4 b^4 + w_5 b^5 = B \quad (12)$$

$$2w_2 + 6w_3 a + 12w_4 a^2 + 20w_5 a^3 = C \quad (13)$$

$$2w_2 + 6w_3 b + 12w_4 b^2 + 20w_5 b^3 = D \quad (14)$$

$$24w_4 + 120w_5 a = E \quad (15)$$

$$24w_4 + 120w_5 b = F \quad (16)$$

After solving this system of six equations with six unknowns, the general form of the polynomial $J(x)$ is achieved. For minimization, the following equation is used

$$E(p) = \sum_{i=1}^{M} \{ \frac{d^6 y_T(x_i,p)}{dx^6} - f(x_i, y_T(x_i,p), \frac{dy_T(x_i,p)}{dx}, \frac{d^2 y_T(x_i,p)}{dx^2} ..... \frac{d^5 y_T(x_i,p)}{dx^5}) \}^2 \quad (17)$$

3.1 Numerical Implementation

In this section, the solution of three example problems with the help of neural network method is presented

3.1.1 Example 1:. Let us shed some light on the sixth-order differential equation:

$$y^{vi}(x) = -6e^x + y(x) \quad 0 < x < 1 \quad (18)$$

subject to boundary condition

$$\begin{aligned} y(0) = 1; \quad y''(0) = -1; \quad y^{iv}(0) = -3 \\ y(1) = 0; \quad y''(1) = -2e; \quad y^{iv}(1) = -4e \end{aligned} \quad (19)$$

The exact solution of above problem is

$$y(x) = (1-x)e^x \quad (20)$$

The trial solution has been obtained as

$$y_T(x,p) = w_o + w_1 a + w_2 a^2 + w_3 a^3 + w_4 a^4 + w_5 a^5 \quad (21)$$

where $w_5 = \frac{3-4e}{120}$, $w_4 = -\frac{1}{8}$, $w_3 = \frac{3-2e}{9}$, $w_2 = -12$, $w_1 = \frac{23e-66}{15}$ and $w_o = 1$. The values of Neural parameters a_i, b_i and c_i, where i represents numbers of nodes, are shown in Table 1. The network has been trained for 10 equidistant points in [0, 1] and compared results between analytical and neural with five nodes fixed in the hidden layer. Comparison between analytical and neural results is outlined in Table 2. Error between two values is also computed here. Analytical and neural results of Example 1, which are obtained are depicted in Figure 1.

Table 2. Error Estimation of Example 1

x	Analytical Solution	Numerical Solution	Error
0.0000000000	1.0000000000	1.0000000000	0.0000000000
0.1000000000	0.9946538300	0.9908587500	0.0037950800
0.2000000000	0.9771222100	0.9699309700	0.0071912400
0.3000000000	0.9449011700	0.9350350300	0.0098661300
0.4000000000	0.8950948200	0.8835115500	0.0115832700
0.5000000000	0.8243606400	0.8121660400	0.0121945900
0.6000000000	0.7288475200	0.7172044500	0.0116430700
0.7000000000	0.6041258100	0.5941605000	0.0099653100
0.8000000000	0.4451081900	0.4378140300	0.0072941500
0.9000000000	0.2459603100	0.2420990000	0.0038613100
1.0000000000	0.0000000000	0.0000000000	0.0000000000

Fig. 1. Graph of exact solution in comparison with the computed solution of Example 1

Table 3. Values of Neural parameters (Example 2), a_i, b_i and c_i with five nodes

i	a_i	b_i	c_i
1	-0.0734690	-0.6258868	-0.2868379
2	0.0151022	-0.6391466	0.5234077
3	-0.0296600	-0.5823448	0.3504931
4	0.0867887	-0.6028480	-0.2325288
5	0.0018519	-0.4779116	0.7076312

3.1.2 Example 2. Consider the following sixth order differential equation:

$$y^{vi}(x) = 6e^{-x} + y^2(x) \quad 0 < x < 1 \quad (22)$$

subject to boundary condition

$$\begin{aligned} y(0) = 1; \quad y''(0) = 1; \quad y^{iv}(0) = 1 \\ y(1) = e; \quad y''(1) = e; \quad y^{iv}(1) = e \end{aligned} \quad (23)$$

The ANN trial solution is written as

$$y_T(x,p) = w_o + w_1 a + w_2 a^2 + w_3 a^3 + w_4 a^4 + w_5 a^5 \quad (24)$$

where $w_5 = \frac{e-1}{120}$, $w_4 = \frac{1}{24}$, $w_3 = \frac{5e-8}{36}$, $w_2 = -12$, $w_1 = \frac{307e-472}{360}$ and $w_o = 1$. The values of Neural parameters a_i, b_i and c_i, where i represents numbers of nodes, are shown in Table 3. The network is trained for 10 equidistant points with five hidden nodes according to the algorithm. In Table 4, the analytical solutions are compared with neural solutions. Also estimated errors are cited in third column of Table 4. Plot of comparison between (analytical results) and (neural results) of Example 2, is shown in Figure 2.

3.1.3 Example 3. The boundary value problem may be consider as follows:

$$y^{vi}(x) = e^x y^2(x) \quad 0 < x < 1 \quad (25)$$

subject to boundary condition

$$\begin{aligned} y(0) = 1; \quad y'(0) = -1; \quad y''(0) = 1 \\ y(1) = e^{-1}; \quad y'(1) = -e^{-1}; \quad y''(1) = e^{-1} \end{aligned} \quad (26)$$

The exact solution of the problem is

$$y(x) = e^{-x} \quad (27)$$

Table 4. Error Estimation of Example 2

x	Analytical Solution	Numerical Solution	Error
0.0	1.00000000	1.00000000	0.00000000
0.1	0.90483742	0.90483742	0.00000000
0.2	0.81873075	0.81873075	0.00000000
0.3	0.74081822	0.74081822	0.00000000
0.4	0.67032005	0.67032005	0.00000000
0.5	0.60653066	0.60653066	0.00000000
0.6	0.54881164	0.54881164	0.00000000
0.7	0.49658530	0.49658530	0.00000000
0.8	0.44932896	0.44932896	0.00000000
0.9	0.40656966	0.40656966	0.00000000
1.0	0.36787944	0.36787944	0.00000000

Fig. 2. Graph of exact solution in comparison with the computed solution of Example 2

Table 5. Values of Neural parameters (Example 3), a_i, b_i and c_i with five nodes

i	a_i	b_i	c_i
1	1.01544450	0.07528380	0.12410432
2	1.23698750	0.07341860	0.13873990
3	0.09458130	0.07306020	0.26252581
4	0.00376860	0.14662330	0.50823108
5	0.09513450	0.07199000	0.17995931

The ANN trial solution is written as

$$y_T(x,p) = w_o + w_1 a + w_2 a^2 + w_3 a^3 + w_4 a^4 + w_5 a^5 \quad (28)$$

where $w_5 = \frac{19e^{-1}-7}{2}$, $w_4 = \frac{17-46e^{-1}}{2}$, $w_3 = \frac{29e^{-1}-11}{2}$, $w_2 = 12$, $w_1 = -1$ and $w_o = 1$. Table 5 gives the values of Neural parameters a_i, b_i and c_i, at different nodes. Ten equidistant points are taken into consideration in the given domain at five hidden nodes. Comparison of analytical and neural results of Example 3 has been shown in Table 6.

It is well known that other numerical methods are usually iterative in nature, where the step size is fixed before the commencement of the computation. After the attainment of the solution if we decide to know what it is between steps, there is no option but to repeat the procedure from initial stages. Neural Network method may be a sort of reliefs we bring to this problem, where this repetition of iterations may be avoided. One can see from the tables and graphs that the trial solution approach improves the accuracy of the results. Lastly, it would not go to amiss to mention that the implemented algorithm is simple, computationally efficient, and straight forward.

4. CONCLUDING REMARKS

In this paper, a stable and efficient feed forward neural network method for sixth-order differential equations with boundary conditions. Numerical results are presented that exhibit the high accuracy of the proposed algorithms. The paper may be concluded on the note that this method is extremely powerful, it may be employed in future calculations by researchs in various filed of engineering. The mian application of the technique in question lies in finding solutions for a higher order of differential eqution.

Table 6. Error Estimation of Example 3

x	Analytical Solution	Numerical Solution	Error
0	1.00000000	1.00000000	0.00000000
0.1	1.10517092	1.10585595	-0.00068503
0.2	1.22140276	1.22270049	-0.00129774
0.3	1.34985881	1.35163868	-0.00177987
0.4	1.49182470	1.49391357	-0.00208887
0.5	1.64872127	1.65091952	-0.00219825
0.6	1.82211880	1.82421682	-0.00209802
0.7	2.01375271	2.01554776	-0.00179505
0.8	2.22554093	2.22685442	-0.00131349
0.9	2.45960311	2.46029828	-0.00069517
1	2.71828183	2.71828183	0.00000000

Fig. 3. Graph of exact solution in comparison with the computed solution of Example 3

5. CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests regarding the publication of this paper.

6. ACKNOWLEDGMENT

We thank the reviewers for their thorough efforts in editing our paper and highly appreciate the comments and constructive criticism that significantly contributed in improving the quality of the publication. The authors also thank Ms. Wishaal Khalid for proofreading our research paper.

7. REFERENCES

[1] Schalkoff, R.J. (1997) *Artificial Neural Networks*. McGraw-Hill, New York.

[2] Picton, P. (2000) *Neural Networks*. 2nd ed. Palgrave, Great Britain.

[3] Rosenblatt, F. (1957) *The perceptron - a perceiving and recognizing automation*. Comell Aeronautical Laboratory, Report 85–460.

[4] Boutayeb, A. & Twizell, E. (1992) *Numerical methods for the solution of special sixth-order boundary value problems*.International Journal of Computer Mathematics,45. pp 207-233

[5] Twizell, E. & Boutayeb, E. (1990) *Numerical methods for the solution of special and general sixth-order boundary value problems, with applications to Benard layer eigenvalue problems*. Proceedings of the Royal Socity London Series A. 431(1883). pp 433-450

[6] Baldwin, P. (1987) *Asymptotic estimates of the eigenvalues of a sixth-order boundary-value problem obtained by using global phase-integral methods*. Philosophical Transactions of the Royal Society of London Series A, 322(1566). pp 281-305

[7] Siddiqi, S.S. & Twizell, E.H. (1996) *Spline solutions of linear sixth-order boundary-value problems*. International Journal of Computer Mathematics, 60(3-4). pp 295-304

[8] El-Gamel, M., Cannon, J.R. & Zayed, A.I. (2004) *Sinc-Galerkin method for solving linear sixth-order boundary-value problems*. Mathematics of Computation, 73(247). pp 1325-1343

[9] Agarwal, R.P. (1986) *Boundary Value Problems for Higher Order Differential Equations*. World Scientific Publishing, Singapore

[10] Shen, J. (1994) *Efficient spectral-Galerkin method. I. Direct solvers of second- and fourth-order equations using Legendre polynomials*. SIAM Journal on Scientific Computing, 15(6). pp 1489-1505

[11] Doha, E.H. & Bhrawy, A.H. (2006) *Efficient spectral-Galerkin algorithms for direct solution for second-order differential equations using Jacobi polynomials*. Numerical Algorithms, 42(2). pp 137-164

[12] Doha, E.H. & Bhrawy, A.H. (2008) *Efficient spectral-Galerkin algorithms for direct solution of fourth-order differential equations using Jacobi polynomials*. Applied Numerical Mathematics, 58(8). pp 1224-1244

[13] Doha, E.H. & Bhrawy, A.H. (2009) *A Jacobi spectral Galerkin method for the integrated forms of fourth-order elliptic differential equations*. Numerical Methods for Partial Differential Equations, 25(3).pp 712-739

[14] Doha, E.H., Bhrawy, A.H. & Abd-Elhameed, W.M. (2009) *Jacobi spectral Galerkin method for elliptic Neumann problems*. Numerical Algorithms, 50(1). pp 67-91

[15] Doha, E.H., Bhrawy, A.H. & Hafez, R.M. (2011) *A Jacobi-Jacobi dual-Petrov-Galerkin method for third- and fifth-order differential equations*. Mathematical and Computer Modelling, 53(9-10). pp 1820-1832

[16] Doha, E.H., Bhrawy, A.H. & Hafez, R.M. (2011) *A Jacobi dual-Petrov-Galerkin method for solving some odd-order ordinary differential equations*. Abstract and Applied Analysis, Artical ID 947230, 21 pages

[17] Bhrawy, A.H. & Abd-Elhameed, W.M. (2011) *New algorithm for the numerical solutions of nonlinear third-order differential equations using Jacobi-Gauss collocation method*. Mathematical Problems in Engineering, Article ID 837218, 14 pages

[18] Doha, E.H., Bhrawy, A.H. & Saker, M.A. (2011) *Integrals of Bernstein polynomials: an application for the solution of high even-order differential equations*. Applied Mathematics Letters, 24(4). pp 559-565

[19] Doha, E.H., Bhrawy, A.H. & Saker, M.A. (2011) *On the derivatives of Bernstein polynomials: an application for the solution of high even-order differential equations*. Boundary Value Problem, Artical ID 829543, 16 pages

[20] Loghmani, G.B. & Ahmadinia, M. (2007) *Numerical solution of sixth order boundary value problems with sixth degree B-spline functions*. Applied Mathematics and Computation, 186(2—). pp 992-999

[21] Siddiqi, S.S. & Akram, G. (2008) *Septic spline solutions of sixth-order boundary value problems*. Journal of Computational and Applied Mathematics, 215(1). pp 288-301

[22] Wazwaz, A. (2001) *The numerical solution of sixth-order boundary value problems by the modified decomposition method*. Applied Mathematics and Computation, 118(2-3). pp 311-325

[23] Bhrawy, A.H. (2009) *Legendre-Galerkin method for sixth-order boundary value problems*. Journal of the Egyptian Mathematical Society, 17(2). pp 173-188

[24] Lee. H. & Kang, I.S. (1990) *Neural algorithm for solving differential equations*. Journal of Computational Physics, 91(1). pp 110-131

[25] Meade, A.J. & Femandez, A.A. (1994) *The numerical solution of linear ordinary differential equations by feedforward neural networks*. Mathematical and Computer Modelling, 19(12). pp 1-25

[26] Meade, A.J. & Femandez, A.A. (1994) *Solution of nonlinear ordinary differential equations by feedforward neural networks*. Mathematical and Computer Modelling, 20(9). pp 19-44

[27] Lagaris, I.E., Likas, A. & Fotiadis, D.I. (1998) *Artificial neural networks for solving ordinary and partial differential equations*. IEEE Transactions on Neural Networks, 9(5) pp 987-1000

[28] Lagaris, I.E., Likas, A. & Papageorgiou, D.G. (2000) *Neural-network methods for boundary value problems with irregular boundaries*. IEEE Transactions on Neural Networks, 11(5). pp 1041-1049

[29] Parisi, D.R., Mariani, M.C. & Laborde, M.A. (2003) *Solving differential equations with unsupervised neural networks*. Chemical Engineering and Processing, 42(8-9). pp 715-721

[30] Malek, A. & Shekari, B. (2006) *Numerical solution for high order differential equations using a hybrid neural network-Optimization method*. Applied Mathematics and Computation, 183(1). pp 260-271

[31] Choi, B. & Lee, J.H. (2009) *Comparison of generalization ability on solving differential equations using backpropagation and reformulated radial basis function networks*. Neurocomputing, 73(1-3). pp 115-118

[32] Yazdi, H.S., Pakdaman, M. & Modaghegh, H. (2011) *Unsupervised kernel least mean square algorithm for solving ordinary differential equations*. Neurocomputing, 74(12-13). pp 2062-2071

[33] Selvaraju, N. & Samant, A. (2010) *Solution of matrix Riccati differential equation for nonlinear singular system using neural networks*. International Journal of Computer Applications, 29. pp 48-54

[34] He, S., Reif, K. & Unbehauen, R. (2000) *Multilayer neural networks for solving a class of partial differential equations*. Neural Networks, 13(3). pp 385-396

[35] Kumar, M. & Yadav, N. (2011) *Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: a survey*. Computers and Mathematics with Applications, 62(10). pp 3796-3811

[36] Tsoulos, I.G., Gavrilis, D. & Glavas, E. (2009) *Solving differential equations with constructed neural networks*. Neurocomputing, 72(10-12). pp 2385-2391

[37] Jianyu, L., Siwei, L, Yingjian, Q. & Yaping, H. (2003) *Numerical solution of elliptic partial differential equation using radial basis function neural networks*. Neural Networks, 16(5-6). pp 729-734

[38] Shirvany, Y., Hayati, M. & Moradian, R. (2009) *Multilayer perceptron neural networks with novel unsupervised training method for numerical solution of the partial differential equations*. Applied Soft Computing Journal, 9(1). ppp 20-29

[39] Mai-Duy, N. & Tran-Cong, T. (2001) *Numerical solution of differential equations using multiquadric radial basis function networks*. Neural Networks, 14(2). pp 185-199

[40] Leephakpreeda, T. (2002) *Novel determination of differential-equation solutions: universal approximation method*. Journal of Computational and Applied Mathematics, 146(2). pp 443-457

[41] Franke, C. & Schaback, R. (1998) *Solving partial differential equations by collocation using radial basis functions*. Applied Mathematics and Computation, 93(1). pp 73-82

[42] Smaoui, N. & Al-Enezi, S. (2004) *Modelling the dynamics of nonlinear partial differential equations using neural networks*. Journal of Computational and Applied Mathematics, 170(1). pp 27-58

[43] Tsoulos, I.G. & Lagaris, I.E. (2006) *Solving differential equations with genetic programming*. Genetic Programming and Evolvable Machines, 7(1). pp 33-54

[44] McFall, K.S. & Mahan, J.R. (2009) *Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions*. IEEE Transactions on Neural Networks, 20(8). pp 1221-1233

[45] Hoda, S.A. & Nagla, H.A. (2011) *Neural network methods for mixed boundary value problems*. International Journal of Nonlinear Science, 11. pp 312-316

[46] Zurada, J.M. (1994) *Introduction to Artificial Neural Network*.West Publishing.

[47] Haykin, S. (1999) *Neural Networks a Comprehensive Foundation*. Prentice Hall, New York.