International Journal of
Science and Engineering Investigations vol. 7, issue 82, November 2018
ISSN: 2251-8843

Artificial Neural Networks: A Meshfree Numerical Method for Solving Engineering Problems Modeled by Differential Equations

Alcineide Pessoa[1], Marta Barreiros[2], Thaís Alves[3], Gean Sousa[4]
[1,2,3]University of Maranhão
[4]Professor at the Federal University of Maranhão
([1]alcineidedutra@hotmail.com, [2]marta-barreiros@hotmail.com, [3]thaishortiz3@gmail.com, [4]gean_mat@yahoo.com.br)

Abstract-Many engineering problems are modeled by Differential Equations and these are solved by some numerical method. The most known numerical methods are dependent on the anticipated choice of a computational mesh. The definition of a mesh can be a constraint when the problem domain under analysis has a very irregular geometry. In the perspective of contributing with propositions about methods that are independent of a computational network, in this article we propose the use of Artificial Neural Networks (RNA). We present the basic RNA concepts and the optimization method that will be used. We apply the network in a second order Ordinary Differential Equation (ODE) that models oscillation problems. We vary the values of the domain and present the results in graphs. We verified that even without a pre-defined mesh RNA obtains a good approximation of the real solution.

Keywords- *Numerical Methods, Differential Equations, Neural Network.s*

I. INTRODUCTION

Several engineering problems can be modeled using Ordinary Differential Equations (EDO) or Partial Differential Equations (EDP) [1]-[3]. To solve such problems computationally some numerical methods have been developed and studied over time. Numerical methods include the Finite Differences Method and the Finite Element Method [4] - [6].

The Finite Differences Method consists a priori of approximations of the derivatives from the Taylor series and the substitution of these approximations into the equation of derivatives to be solved [4]. Through this method it is possible to obtain solutions for a discrete domain known as computational mesh. This mesh is composed of a finite number of rectangles, and on the vertices of them the solution of the equation is computed. The solutions obtained by this method are usually stable if and only if the mesh is chosen under some stability criterion. If this method is applied to non-uniform geometries, the computational mesh does not serve the entire problem domain [7], [8].

In order to solve the problem of approximation of irregular geometries related to the Finite Differences Method, the Finite Element Method arises. In this method it is possible to refine the computational mesh using not only rectangles, but triangles, hexagons etc. [5]. Even with the increase in the possibility of choosing the element to be used in the creation of the mesh, the Finite Element Method can also have problems of instabilities in the solutions, caused by the choice of mesh elements [9,10]. In view of the foregoing, it is clear that proposing methods capable of solving differential equations without dependence on a predefined mesh element is an important contribution to the field of numerical methods applied in engineering.

On the other hand, Artificial Neural Networks (ANNs) have been widely used in engineering problems, especially for predicting difficult-to-measure variables. In the perspective of collaborating with the construction of methods capable of solving mathematical models applied to engineering, without the dependence of pre-established computational mesh is that in this article we propose the use of ANNs to solve a second order ODE [8].

The mathematical construction of the proposal is based on RNA concepts and optimization methods in the literature, especially the Descending Gradient Method. For example, a computational simulation is presented in order to present the numerical results of the proposed problem.

II. ARTIFICIAL NEURAL NETWORKS

Constructing mathematical models capable of simulating some biological functions has been a challenge for researchers in the exact sciences. This challenge is a motivator for scientists working in the large area of knowledge known as artificial intelligence to develop neural models capable of learning patterns. These models are usually used in problems of classification, forecasting, etc. [11].

A. Biological Motivation

The mathematical construction of ANNs is based on the electrical, chemical and biological relationships that occur in the human nervous system. In this system the importance of the neurons stands out. Neurons are excitable (or self-excitable) cells that communicate with each other through synapses, forming functional networks for information processing and storage [12]. The main function of neurons is to conduct electrical stimuli from physical-chemical reactions. Three main parts constitute a neuron: dendrites, cell body (soma) and axon. In a brief way it can be defined that the dendrites are receptors of information, the synaptic input. The cell body contains the nucleus of the cell, where all the hormones, proteins and neurotransmitters are produced. Finally, axons, whose endings make synaptic contacts with other neurons [12]. (Fig. 1)

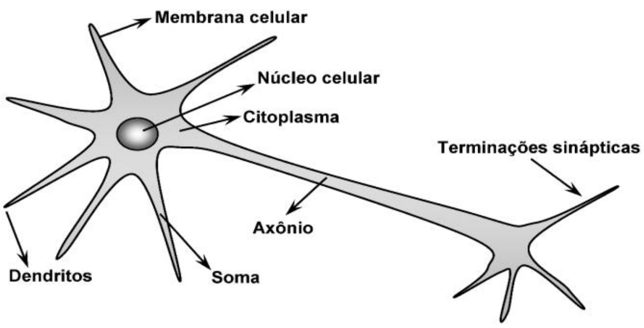

Figure 1. Biological Neuron Model [11].

B. Artificial Neuron

Based on the functioning of the biological neurons McCulloch & Pitts (1943) developed a mathematical model capable of encompassing the main characteristics of the biological neuron [13]. This model, although simple, is still the most used model in the various ANN architectures [11].

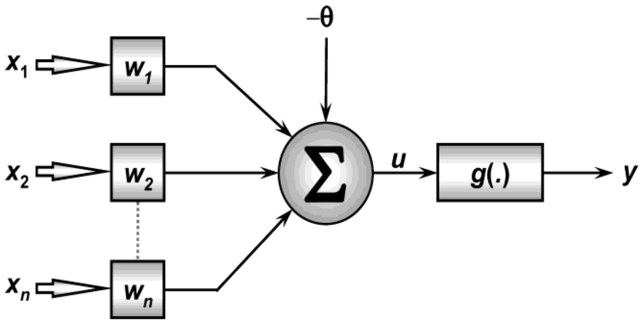

Figure 2. Artificial neuron model [11].

The neural model of McCulloch & Pitts (1943) consists of seven basic elements [11]: input information $\{x_1, x_2, \ldots, x_n\}$; synaptic weights $\{w_1, w_2, \ldots, w_n\}$; linear combiner $\sum$; Activation threshold θ (also known as "bias"); activation potential u (difference between the result of the linear combination and the activation threshold); the activation function g and the output signal y.

C. Perceptron Multi-layered and Backpropagation Algorithm

In an artificial neural network neurons are organized in the form of layers, and the way these layers are arranged defines the network architecture [14]. In this article, in particular, a network known as Perceptron Multi-layered will be used. Perceptron is the simplest form of a neural network used initially for classification of linearly separable patterns. This network consists of a single neuron with adjustable synaptic weights [12]. The Perceptron of Multiple Layers (PMC) consists of a neural structure composed of a layer of input neurons, one or more hidden layers and an output layer [15].

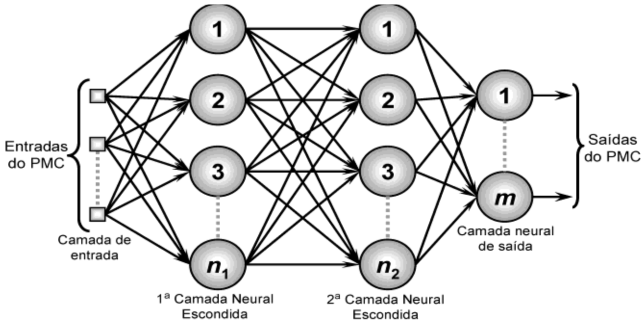

Figure 3. Perceptron with two hidden neural layers [11].

The most common training algorithm of a PMC is Backpropagation (retro propagation). This algorithm consists in the use of the network output errors to update the synaptic weights in a retroactive way, that is, "from the output to the input network" [16]. The back propagation algorithm is implemented considering two phases, forward and backward. In the forward phase the input values are multiplied by the synaptic weights in the input/output direction, at the end of that phase an error is calculated. Already in the backward phase, optimization techniques are applied on the error in such a way that the synaptic weights are adjusted in the output / input order [10].

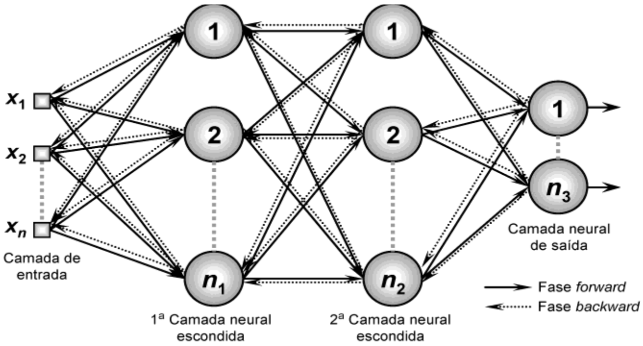

Figure 4. Retro propagation algorithm [11].

III. METHODOLOGY

A. Update of the Synaptic Weights

Solving an ODE using an RNA means finding an algorithm that minimizes (or maximizes) a cost function. This cost function must be chosen in such a way that its minimization (or maximization) is equivalent to solving ODE. Considering that a second order is an equation of type [15]:

$$\frac{d^2\psi(x)}{dx^2} = \left(f(x, \psi, \frac{d\psi}{dx}\right) \quad (1)$$

then the approximate solution can be written in the form: $\psi_t(x) = A + A^{'}x + x^2 N(x,p)$

with the initial values given by $\psi(0) = A$ e $\frac{d\psi(0)}{dx} = A1$. Consequently, the cost function to be minimized is $E[\vec{p}] = \sum_i \left\{\frac{d^2\psi_t(X_i)}{dx^2} - f(x_i, \psi_t, \frac{d\psi_t(x_i)}{dx}\right\}^2$ where xi are the points of the problem domain. Considering that the RNA model we will use is composed of two layers, an input layer and a hidden layer and that w_1 is the vector of weights of the input layer and w_2 the weights of the hidden layer, then by the optimization method known as Descending Gradient [17].

$$w_1(n+1) = w_1(n) - \mu \frac{\partial E[\vec{p}]}{\partial w_{ki}} \quad (2)$$

and

$$w_2(n+1) = w_2(n) - \mu \frac{\partial E[\vec{p}]}{\partial v} \quad (3)$$

If we start the values of $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ at random then by the given conditions the method will converge to the minimum of the cost function presented [15].

Solving derivatives $\frac{\partial E[\vec{p}]}{\partial w_{ki}}$ and $\frac{\partial E[\vec{p}]}{\partial v_k}$ we obtain:

$$\frac{\partial E[\vec{p}]}{\partial v_k} = 2y \left\{ \begin{array}{l} 2\ \frac{\partial N(x_1,\vec{p}\ )}{\partial v_k} + 4x_i \frac{\partial \frac{dN(x_i,\vec{p})}{dx_i}}{\partial v_k} + \\ x_i^{2\frac{\partial \frac{d^2N(x_i,\vec{p})}{dx_i^2}}{\partial v_k}} - \frac{\partial f(x_i, \psi_t(x_i), \frac{d\,\psi_t(x_i)}{dx}}{\partial v_k} \end{array} \right\} \quad (4)$$

Where

$$y = 2N\ (x_1, \vec{p}) + 4x_1 \frac{dN(x_1,\vec{p})}{dx_i} + x_1^2 \frac{d^2N(x_1,\vec{p})}{{dx_i}^2} - -f\left(x_1, \psi_t(x_1), \frac{d\,\psi_t(x_i)}{dx}\right) \quad (5)$$

The gradient of the cost function relative to the weights of the hidden layer is:

$$\frac{\partial E[\vec{p}]}{\partial w_{ki}} = 2y \Big\{ 2v_k\sigma_k' x_i + 4x_i[v_k\sigma_k' + v_k w_{ki}\sigma_k'' x_i] + x_i^2\left(2v_k w_{ki}\sigma_k'' + v_k\, w_{kj}^2 \sigma_k''' x_i\right) - - \frac{\partial f(x_i, \psi_t(x_i), \frac{d\,\psi_t(x_i)}{dx}}{\partial \psi_t(x_{i)}}, x_i^2 v_k \sigma_k' x_i \Big\} \quad (6)$$

More details on the calculation of the presented gradients can be found in [15].

IV. SIMULATIONS

The neural model presented previously combined with their respective synaptic weights update algorithms will be applied to a second order problem in order to verify the behavior of the solutions when choosing a domain with random points. The problem chosen to support simulation is well known in civil engineering and consists of an ODE that models the vertical movement of the Tacoma bridge [18]:

$$\frac{d^2y}{dt^2} = -b\frac{dy}{dt} - cy + w(t), \quad (7)$$

Where

- w: function that models the wind force;
- b damping coefficient;
- c measure the resistance of the cables.

It is worth mentioning that the model presented is only one of several proposals for modeling this problem. We will only use it as an illustration to verify the RNA's ability to solve this type of problem without depending on a pre-established computational mesh.

V. RESULT

Before presenting the simulation parameters we will make the following assumptions:

a) The damping coefficient b varies with time $b = \frac{6}{3t} + 2$;

b) The resistance of the cables c also varies with time $c = \frac{1}{t} + 1$;

c) The function $w(t) = t^2 - 2$;

d) The initial conditions $y(0) = 0$ and $dy\frac{(0)}{dt} = 0$, are satisfied.

e) The parameters used in the neural network were as follows:

f) An input layer and a hidden layer with weights $w1$ and $w2$;

g) Number of neurons in the hidden layer equal to the number of neurons in the input layer equal to 5;

h) Sigmoid activation function;

i) Learning rate 0.5

In order to compare the results, all synaptic weights were started with the same values

$$w_1 = |0.157 \quad 0.970 \quad 0.957 \quad 0.485 \quad 0.800|$$

$$w_2 = |0.141 \quad 0.421 \quad 0.915 \quad 0.792 \quad 0.959|$$

More details on the calculation of the presented gradients can be found in [15].

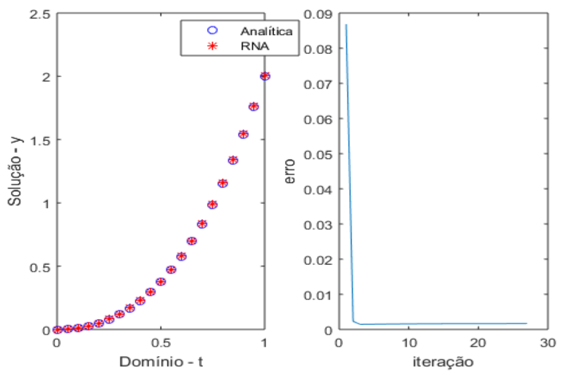

Figure 5. Results (solution and error) for a domain with equally spaced points.

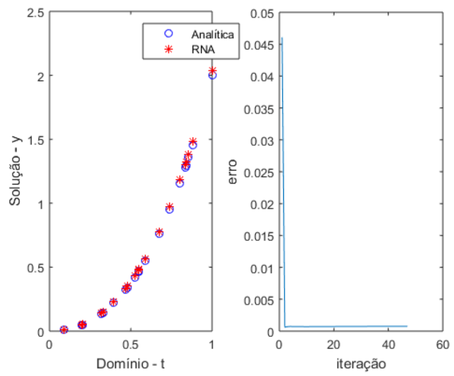

Fig. 8- results (solution and error) for a domain with randomly spaced points

Figures 6,7,8 are comparisons between the analytical solution and that obtained by the neural network when the domain was determined by choosing points in the interval [0,1] in a random manner. It is important to note that comparisons with methods such as Finite Differences and Finite Elements have not been made because these traditional methods depend on a pre-defined mesh and thus it is not possible to obtain solutions in a domain where the points are chosen at random.

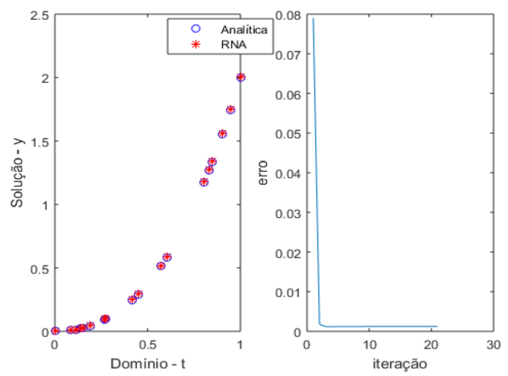

Figure 6. Results (solution and error) for a domain with randomly spaced points.

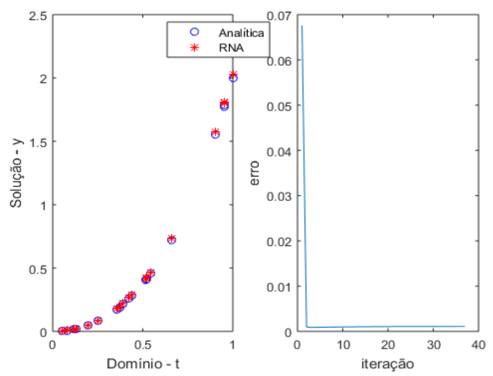

Figure 7. Results (solution and error) for a domain with randomly spaced points

VI. Discussion

Observing the graphs presented in the results it is possible to make some observations. Firstly, the best result was obtained when the domain points were equally spaced (Fig. 5). On the other hand, when the domain points were randomly selected RNA was able to present solutions very close to the analytical solutions (Fig. 6-8). This fact is a motivator for the continuation of the research with applications in other problems and consequently other models of equations.

If the effectiveness of RNA is solving problems, regardless of how the domain is chosen, is confirmed for other problem then it will be possible to use this numerical method as an alternative to be used in problems whose geometry under analysis is a limitation for traditional methods.

It is still verified in the graphs referring to the behavior of the error in each iteration that there is a fast convergence of the method because in all cases the error is close to zero with less than 10 iterations (Fig. 6-8).

VII. Conclusion

The model of Artificial Neural Networks applied to the problem modeled by EDO was efficient when applied in a mesh of equally spaced points. This model when used in a random point domain, obtained promising results. It is worth stressing the importance of the emphasis on domain choice, since the convergence of traditional methods such as Finite Differences and Finite Elements is entirely dependent on the point mesh chosen as domain of the problem.

The proposal presented here is only a starting point for the improvement and development of similar methodologies that are capable of solving problems modeled by EDO'S and EDP's such as the problem of heat propagation, wave propagation, Timoshenko and so on.

References

[1] K. S. Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," IEEE Transactions on Antennas and Propagation. 1966.

[2] B. Engquist and A. Majda, "Absorbing boundary conditions for the numerical simulation of waves," Math. Comput., 1977.

[3] Z.-H. Huang and T.-Z. Huang, "a Constraint Preconditioner for Solving Symmetric Positive Definite Systems and Application To the Helmholtz Equations and Poisson Equations," Math. Model. Anal., vol. 15, no. 3, pp. 299–311, 2010.

[4] O. Adekanye and T. Washington, "Nonstandard finite difference scheme for a Tacoma Narrows Bridge model," Appl. Math. Model., 2018.

[5] J. Bai, Y. Cao, X. He, H. Liu, and X. Yang, "Modeling and an immersed finite element method for an interface wave equation," Comput. Math. with Appl., vol. 76, no. 7, pp. 1625–1638, 2018.

[6] A. Oulhaj, S. Al Dhaheri, B. Bin Su, and M. Al-Houqani, "Discriminating between positional and non-positional obstructive sleep apnea using some clinical characteristics," Sleep Breath., 2017.

[7] J. C. Butcher, Numerical Methods for Ordinary Differential Equations. 2016.

[8] R. E. Mickens, "Calculation of denominator functions for nonstandard finite difference schemes for differential equations satisfying a positivity condition," Numer. Methods Partial Differ. Equ., 2007.

[9] W. Guo, Y. Nie, and W. Zhang, "Parallel adaptive mesh refinement method based on bubble-type local mesh generation," J. Parallel Distrib. Comput., vol. 117, pp. 37–49, 2018.

[10] X. Tang, L. Zou, W. Yang, Y. Huang, and H. Wang, "Novel mathematical modelling methods of comprehensive mesh stiffness for spur and helical gears," Appl. Math. Model., 2018.

[11] F. A. Gomide, "Redes neurais artificiais para engenharia e ciências aplicadas: curso prático," Sba Control. Automação Soc. Bras. Autom., 2012.

[12] L. Fleck, M. H. F. Tavares, E. Eyng, and A. HELMANN, "Redes Neurais Artificiais: Princípios Básicos," Rev. Eletrônica Científica Inovação e Tecnol., 2016.

[13] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bull. Math. Biophys., 1943.

[14] D. Reby, S. Lek, I. Dimopoulos, J. Joachim, J. Lauga, and S. Aulagnier, "Artificial neural networks as a classification method in the behavioural sciences," Behav. Processes, vol. 40, no. 1, pp. 35–43, 1997.

[15] B. Batista, "Soluções de Equações Diferenciais Usando Redes Neurais de Múltiplas camadas com os métodos da Descida mais íngreme e Levenberg-Marquardt.," Ppgme.Ufpa.Br, 2012.

[16] H. Yan, Y. Jiang, J. Zheng, C. Peng, and Q. Li, "A multilayer perceptron-based medical decision support system for heart disease diagnosis," Expert Syst. Appl., 2006.

[17] Y. Yuan, "Step-sizes for the gradient method," Ams Ip Stud. Adv. Math., 2008.

[18] B. A. Fillenwarth and T. M. Washington, "Improving the Mathematical Model of the Tacoma Narrows Bridge," vol. 8, no. 2, 2007.

Alcineide Dultra Pessoa, Bachelor in science and technology and major in Civil Engineering at the Federal University of Maranhão.

Marta de Oliveira Barreiros, Bachelor in Information Systems from the Faculty of Imperatriz, Master in Biotechnology from the Federal University of Pará. PhD student in the Graduate Program in Electrical Engineering, Federal University of Maranhão.

Thaís Hortiz Santos Alves, Graduated in Science and Technology from the Federal University of Maranhão.

Gean Carlos Lopes de Sousa, Graduated and Masters in Mathematics from the Federal University of Pará. He is currently a Professor at the Federal University of Maranhão.