Research Article

Ahmad Jafarian* and Dumitru Baleanu

# Application of ANNs approach for wave-like and heat-like equations

https://doi.org/10.1515/phys-2017-0135
Received Aug 31, 2016; accepted Jun 09, 2017

**Abstract:** Artificial neural networks are data processing systems which originate from human brain tissue studies. The remarkable abilities of these networks help us to derive desired results from complicated raw data. In this study, we intend to duplicate an efficient iterative method to the numerical solution of two famous partial differential equations, namely the wave-like and heat-like problems. It should be noted that many physical phenomena such as coupling currents in a flat multi-strand two-layer super conducting cable, non-homogeneous elastic waves in soils and earthquake stresses, are described by initial-boundary value wave and heat partial differential equations with variable coefficients. To the numerical solution of these equations, a combination of the power series method and artificial neural networks approach, is used to seek an appropriate bivariate polynomial solution of the mentioned initial-boundary value problem. Finally, several computer simulations confirmed the theoretical results and demonstrating applicability of the method.

**Keywords:** wave-like and heat-like equations; bivariate power series polynomial; artificial neural network; criterion function; back-propagation learning algorithm

**PACS:** 07.05.Mh; 02.60.Jh

## 1 Introduction

As known, differential equations occur in many science phenomena. Whenever, there is a meaningful relation between different values, states or times in which the rate of variable changes at different times or states are known, it can be modeled via differential equations. Hence, various applications of differential equations have led mathematicians and engineer scientist to focus on appropriate methods in solving these kinds of mathematical problems. More complicated differential equations arising from the modeling of complex phenomena can not be solved simply by existing conventional methods. Therefore, finding alternative numerical techniques seems to be necessary. During recent decades, various numerical techniques have been used for approximating different types of differential equations in linear and nonlinear cases. Among these, we can mention the homotopy perturbation method [2, 4, 5], variational iteration method [21, 22, 26], Adomian decomposition method [1, 3, 10], etc. An effective method in solving a wide variety of complicated mathematical problems is artificial neural networks (ANNs) approach. Since, these networks have high efficiency in approximating solutions of mathematical problems. Recently, some structures of neural networks have been applied for solving variety of mathematical problems [11–13].

In this paper, a suitable structure of neural networks will be applied for solving initial-boundary one-dimensional wave-like and heat-like equations. The heat-like equation can be a mathematical modeling for temperature changes of the composite materials. The solution of the mentioned partial differential equation is proportional to kinetic energy of particles in the material. Meanwhile, the wave-like equation can describe moving the particles under Hook's law in the disordered systems. In this kind of equation, the standard initial and boundary conditions can be considered. In recent years, some researchers have solved numerically these problems [14, 18–20, 24, 25]. To do this, a four-layer feed-forward neural network corresponding to the mentioned bivariate power series solution is designed satisfying both initial and boundary conditions. The designed neural architecture can easily approximate solution functions on the space of unknown constant coefficients using a suitable learning algorithm. In other words, after discretizing the differential domain using a standard rule and doing some simplifications, the origin problem is transformed to solve a minimizing optimization problem. The yield nonlinear problem is solved iteratively using a

***Corresponding Author: Ahmad Jafarian:** Department of Mathematics, Urmia Branch, Islamic Azad University, Urmia, Iran; Email: jafarian5594@yahoo.com
**Dumitru Baleanu:** Department of Mathematics and Computer Sciences, Faculty of Art and Sciences, Cankaya University, 06530 Balgat, Ankara, Turkey; Institute of Space Sciences, Magurele-Bucharest, Romania; Email: dumitru@cankaya.edu.tr

Open Access. © 2017 A. Jafarian and D. Baleanu, published by De Gruyter Open. BY-NC-ND This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License

Brought to you by | Cankaya University
Authenticated
Download Date | 12/18/19 11:48 AM

back propagation learning rule, which is based on the gradient descent method. Approximating the constant series coefficients lead to finding the mentioned series solution on the given differential domain. In Section 2, we will first have an overview of artificial neural nets and their computational process. Then, having the introduction of the proposed neural architecture, the numerical solution of the mentioned differential equations will be considered using the defined combination iterative method. Two numerical examples with computer simulations are presented in Section 3. Also, to better express the effectiveness of the presented technique, the obtained numerical results will compare via the ones achieved from another classical method. Finally, conclusions and recommendation for future research are presented in Section 4.

## 2 Description of the method

The general purpose of this section is to introduce a combination iterative method for approximating solution of two well known partial differential equations. In order to better clarify all the fundamental mathematical features of the method presented in this research, we first deal with more general theories of neural networks issue. The proposed iterative technique which is based on the combination of power series method and a modification of ANNs approach, is used to find approximate solutions of boundary-initial value wave-like and heat-like equations on a given closed domain.

### 2.1 Basic idea of ANNs

Artificial neural networks theory revolves around the idea that certain key processing properties of human being brain can be modeled and applied to approximate computational methods using biological processes. In other words, artificial neural networks attempt to get knowledge of the relation between a set of data through training and finally store the knowledge gained for similar purpose. The main ideas of these networks are partly inspired by a way biological nervous system functions, to process data and information for learning and createing knowledge. In a neural network, simple processing elements are known as "neurons", which can display complex global behavior determined by the connection between processing elements and element parameters. A neural net can not be adaptive itself. The practical application of this network is needed to enable employing algorithms which are designed to alter and adjust unknown parameters. For this purpose, using knowledge of computer programming, we can design a structure that acts as a nervous system. So, a learning algorithm is defined for networks by creating a network of interconnected artificial neurons. Mentioned networks have exhibited high performance on estimation and approximation. There are many references on neural nets field, see for example Ref. [6–8].

Now, let's consider the neural architecture shown in Figure 1. This network is a four-layer feed-forward neural architecture with two input signals and one output neuron. Each input signal is multiplied by its respective weight values, and these weighted signals are then perfectly summed to produce the next layer's inputs. In other words, each hidden or output layer receives its inputs from the previous weighted neurons and then presents it to a suitable activation function. The network output can be calculated. The remarkable notation in this area is that the bias term in output layer is added with the weighted signals of second hidden layer and then makes up the overall input of the final layer. According to the above, each neuron's input-output relation can be written as follows:

- *Input units:*

$$o_1^1 = x, \qquad (1)$$

$$o_2^1 = t.$$

- *First hidden units:*

$$o_{1,i}^2 = (o_1^1)^i, \; i = 1, \dots, n, \qquad (2)$$

$$o_{2,j}^2 = (o_2^1)^j, \; j = 1, \dots, m.$$

- *Second hidden units:*

$$o_{i,j}^3 = o_{1,i}^2 . o_{2,j}^2, \; i = 1, \dots, n, \; j = 1, \dots, m. \qquad (3)$$

- *Output unit:*

$$N(x,t) = \sum_{i=1}^{n}\sum_{j=1}^{m}(a_{i,j}.o_{i,j}^3) + \sum_{i=1}^{n}(a_{i,0}.o_{1,i}^2) + \sum_{j=1}^{m}(a_{0,j}.o_{2,j}^2) + a_{0,0}. \qquad (4)$$

The designed neural structure is a prototype model, which should be created having some minor changes caused by conditions of problem. In other words, by presenting boundary or initial conditions of a given partial differential equation problem, the neural architecture will be ready to learn. This will cause the network output desirability to the solution of the mentioned math problem.

Brought to you by | Cankaya University
Authenticated
Download Date | 12/18/19 11:48 AM

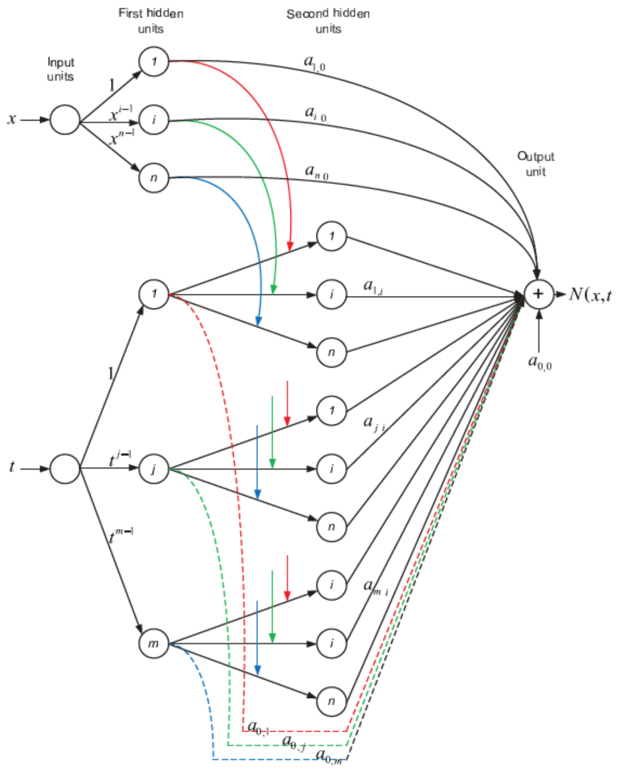

Figure 1: The designed neural network architecture

## 2.2 Implementation of the method

As mentioned before, the main idea of this study is to apply the designed multi-layer feed-forward neural architecture to the function approximation of two famous types of partial differential equations. Consider the one-dimensional wave-like Equation (5) and heat-like Equation (6):

$$\frac{\partial u}{\partial t^2} - k(x,t)\frac{\partial^2 u}{\partial x^2} = h(x,t),\ 0 < x < 1,\ t > 0, \tag{5}$$

$$\frac{\partial u}{\partial t} - k(x,t)\frac{\partial^2 u}{\partial x^2} = h(x,t),\ 0 < x < 1,\ t > 0, \tag{6}$$

subject to the initial conditions:

$$u(x,0) = g_0(x),\ \frac{\partial u(x,t)}{\partial t}|_{t=0} = g_1(x),$$

and the boundary conditions:

$$u(0,t) = f_0(t),\ u(1,t) = f_1(t).$$

In the above relations, $k$ and $h$ are given continuous positive and real-valued functions, respectively. The defined wave-like and heat-like equations are more applicable in modeling different physical phenomena. The main problem occurs when we encounter more complicated modeled problems of these types, in which the existing analytical or numerical methods can not solve them. Therefore, finding an effective alternative method seems to be necessary. In the last decade, some modifications of the power series method have been employed for solving several types of partial differential equations with initial or boundary conditions [16, 17, 23]. Extensive studies on these polynomials, show that this series solution technique is really a powerful tool for solving complex mathematical problems. As previously mentioned, a combination of these polynomials and ANNs approach will be considered as an iterative method for solving the above partial differential equations. The basic idea in this issue is that the solution function $u(x,t)$ on domain $\Omega = [0,1]\times[0,T]$, can be completely represented in a polynomial series of degree $(n,m)$ as:

$$u_{n,m}(x,t) = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,j}x^i t^j, \tag{7}$$

for the constant coefficients $a_{i,j}$ (for $i = 0,\dots,n;\ j = 0,\dots,m$). It should be noted that the solution function can be represented as the power series (7) if and only if it is complex differentiable in the open set $(0,1)\times(0,T)$. It is reasonable to consider that the designed neural architecture are fully associated with the polynomial series (7). The interesting point in this approach is that the growing degree of basis polynomials, increases the accuracy of the recommended combination method. It must be considered that, this work will lead to more complex relations.

### 2.2.1 Discretization of the problems

Power series method is based on the fact that any modification of these series before being used, must be satisfied in the initial or boundary conditions. For the introduced partial differential equations, the trial solution is written as follows:

$$\tilde{u}(x,t) = A(x,t) + x(1-x)t(N(x,t) - N(x,0)), \tag{8}$$

where

$$A(x,t) = (1-x)f_0(t) + xf_1(t) + g_0(x) - \left[(1-x)g_0(0) \right.$$
$$\left. + xg_0(1)\right] + t\{g_1(x) - [(1-x)g_1(0) + xg_1(1)]\},$$

in which the function $A(x,t)$ has been chosen in a manner that satisfies in the both initial and boundary conditions, simultaneously. The introduced trial function involves the given feed-forward architecture that satisfies in both initial and boundary conditions. Supposedly $\tilde{u}(x,t)$ symbolizes the trial solution with adjustable parameter $a_{i,j}$, the problem is transformed from the original construction to

Brought to you by | Cankaya University
Authenticated
Download Date | 12/18/19 11:48 AM

an unconstrained one by direct substitution (8) in the primary equations. So, the Eqs. (5) and (6) are shape changes into the form (9) and (10), respectively:

$$\tilde{u}_{tt}(x,t) - k(x,t)\tilde{u}_{xx}(x,t) = h(x,t),\ (x,t) \in \Omega, \quad (9)$$

$$\tilde{u}_{t}(x,t) - k(x,t)\tilde{u}_{xx}(x,t) = h(x,t),\ (x,t) \in \Omega, \quad (10)$$

where

$$\begin{aligned}\tilde{u}_{xx}(x,t) = A_{xx}(x,t) + t(&-2(N(x,t) - N(x,0)) \\ &+ (2-4x)(N_x(x,t) - N_x(x,0)) \\ &+ (x-x^2)(N_{xx}(x,t) - N_{xx}(x,0))),\end{aligned}$$

$$\tilde{u}_t(x,t) = A_t(x,t) + x(1-x)\{N(x,t) - N(x,0) + tN_t(x,t)\},$$
$$\tilde{u}_{tt}(x,t) = A_{tt}(x,t) + x(1-x)(2N_t(x,t) + tN_{tt}(x,t)),$$

and

$$N_t(x,t) = \sum_{i=0}^{n}\sum_{j=1}^{m} j a_{i,j} x^i t^{j-1},$$
$$N_{tt}(x,t) = \sum_{i=0}^{n}\sum_{j=2}^{m} j(j-1) a_{i,j} x^i t^{j-2},$$
$$N_x(x,t) = \sum_{i=1}^{n}\sum_{j=0}^{m} i a_{i,j} x^{i-1} t^{j},$$
$$N_{xx}(x,t) = \sum_{i=2}^{n}\sum_{j=0}^{m} i(i-1) a_{i,j} x^{i-2} t^{j},$$

$$\begin{aligned}A_t(x,t) = (1-x)f_0'(t) + xf_1'(t) + g_1(x) \\ - ((1-x)g_1(0) + xg_1(1)),\end{aligned}$$

$$A_{tt}(x,t) = (1-x)f_0''(t) + xf_1''(t),$$

$$A_{xx}(x,t) = g_0''(x) + tg_1''(x).$$

Now, we intend to define a set of acceptable mesh points for the discretization of Equations. (9) and (10). For positive integers $n'$ and $m'$, let $\Omega_{p,q}$ be a partition of square $\Omega$ with the mesh points $(x_p, t_q) = (\frac{p}{n'}, \frac{qT}{m'})$, (for $p = 0, \ldots, n'$; $q = 0, \ldots, m'$). Substituting collocation point $(x_p, t_q)$ into the resulted relations, reduces the problems into the following systems of equations:

$$\begin{aligned}&\tilde{u}_{tt}(x_p,t_q) - k(x_p,t_q)\tilde{u}_{xx}(x_p,t_q) = \\ &h(x_p,t_q),\ p = 0,\ldots,n',\end{aligned} \quad (11)$$

$$\begin{aligned}&\tilde{u}_{t}(x_p,t_q) - k(x_p,t_q)\tilde{u}_{xx}(x_p,t_q) \\ &= h(x_p,t_q),\ q = 0,\ldots,m'.\end{aligned} \quad (12)$$

Continuing, we intend to construct an iterative procedure with the help of artificial neural networks approach to solve the resulting systems.

### 2.2.2 Proposed error function

As it is known, in iterative methods a valid criterion is required to measure the produced error of each iteration. Here, the differentiable least mean square (LMS) function is employed to measure the accuracy of solutions. This rule is stated for wave-like and heat-like equations, respectively as follows:

$$\begin{aligned}EW_{p,q} = \frac{1}{2}\big(&\tilde{u}_{tt}(x_p,t_q) - k(x_p,t_q)\tilde{u}_{xx}(x_p,t_q) \\ &-h(x_p,t_q)\big)^2,\end{aligned} \quad (13)$$

$$\begin{aligned}EH_{p,q} = \frac{1}{2}\big(&\tilde{u}_{t}(x_p,t_q) - k(x_p,t_q)\tilde{u}_{xx}(x_p,t_q) \\ &-h(x_p,t_q)\big)^2.\end{aligned} \quad (14)$$

Minimizing the defined error functions over the space of possible weight parameters can be an interested issue. To do this, a set of training rules is build to minimize $EW$ and $EH$ by adaptively adjusting the network parameters. Hence, one suitable error correction technique must be essentially employed to achieve this goal. More details concerning minimizing techniques can be found in Ref. [9].

### 2.2.3 Proposed learning algorithm

What makes this particular use of neural networks so attractive in many applications is that they express the ability to learn, though this remarkable property might be challenged by some researchers. In terms of ANNs, "learning" simply means changing the weights and biases of the network in response to some input data. Once a particular learning algorithm succeeds, programming the network to have a particular unequivocal performance is not vitally important. In other words, we need no prior knowledge to adjust the weights and biases. The designed neural architecture adjusts its parameters for a learning algorithm. That is, the error alters as the weights and bias term are changed. In this sense, the neural network learns from experience. Therefore, the back-propagation algorithm is the most widely-used method for feed-forward networks. The learning rules are mathematical formalizations that are believed to be more effective in the ANN's performance. To fine-tune the neural network, the network parameters are first quantified with arbitrary initial guesses; then, the neural network calculates the output for each input signal. Next, the defined error rule is employed by substituting the proposed network model instead of the solution function in the origin problem. To train the present network, we have employed an optimization technique that in turn

Brought to you by | Cankaya University
Authenticated
Download Date | 12/18/19 11:48 AM

required the computation of the gradient of the error with respect to the net parameters.
Now, a suitable error correction rule must be initially used for single units training. This rule essentially drives the output error of the network to zero. We start with the classical generalized delta learning rule and give a brief description for its performance. Throughout this section, an attempt is made to point out the criterion function that is minimized by using this rule. Learning in neural nets is appropriate selecting the connection weights, which yields to minimize the error function on a set of mesh points. During the training, the initial parameters $a_{i,j}$ are put into the network and flow through the network generating a real value on the output unit. As seen in the last part, the calculated output is compared with the desired one, and an error is computed. The differentiable cost functions $EW$ and $EH$ are always decreasing in the opposite direction of its derivative. It means that if we want to find one of the local minima of this function starting from a initial guess. We employ the supervised back-propagation learning algorithm to reach this goal. The mentioned self learning mechanism starts with randomly quantifying the initial parameters $a_{i,j}$ (for $i = 0, \dots, n;\ j = 0, \dots, m$). The mentioned algorithm is well presented for wave-like equation as follows:

$$a_{i,j}(r+1) = a_{i,j}(r) + \Delta a_{i,j}(r), \tag{15}$$

$$\Delta a_{i,j}(r) = -\eta . \frac{\partial EW_{p,q}}{\partial a_{i,j}} + \gamma . \Delta a_{i,j}(r-1),$$
$$p = 0, \dots, n';\ q = 0, \dots, m',$$

where $\eta$ and $\gamma$ are the learning rate and momentum term, respectively. In the above, the index $r$ in $a_{i,j}(r)$ ascribes to the repetition number and the subscript $i, j$ in $a_{i,j}$ is the label of the training connection weight. Moreover, $a_{i,j}(r+1)$ and $a_{i,j}(r)$ depict the adjusted and current weight parameter, respectively. To complete the derivation of learning procedure for the output layer weights, the above partial derivative can be expressed as follows:

$$\frac{\partial EW_{p,q}}{\partial a_{i,j}} = (\tilde{u}_{tt}(x_p, t_q) - k(x_p, t_q)\tilde{u}_{xx}(x_p, t_q) - h(x_p, t_q)) \times \left( \frac{\partial \tilde{u}_{tt}(x_p, t_q)}{\partial a_{i,j}} - k(x_p, t_q) \frac{\partial \tilde{u}_{xx}(x_p, t_q)}{\partial a_{i,j}} \right),$$

where

$$\frac{\partial \tilde{u}_{tt}(x_p, t_q)}{\partial a_{i,j}} = \frac{\partial A_{tt}(x_p, t_q)}{\partial a_{i,j}} + x(1-x)\left( 2\frac{\partial N_t(x_p, t_q)}{\partial a_{i,j}} + t\frac{\partial N_{tt}(x_p, t_q)}{\partial a_{i,j}} \right),$$

$$\begin{aligned}\frac{\partial \tilde{u}_{xx}(x_p, t_q)}{\partial a_{i,j}} = \frac{\partial A_{tt}(x_p, t_q)}{\partial a_{i,j}} &+ t_q \left( -2 \left( \frac{\partial N(x_p, t_q)}{\partial a_{i,j}} - \frac{\partial N(x_p, 0)}{\partial a_{i,j}} \right) \right. \\ &+ (2 - 4x_p) \left( \frac{\partial N_x(x_p, t_q)}{\partial a_{i,j}} - \frac{\partial N_x(x_p, 0)}{\partial a_{i,j}} \right) \\ &\left. + (x_p - x_p^2) \left( \frac{\partial N_{xx}(x_p, t_q)}{\partial a_{i,j}} - \frac{\partial N_{xx}(x_p, 0)}{\partial a_{i,j}} \right) \right),\end{aligned}$$

and

$$\frac{\partial N(x_p, t_q)}{\partial a_{i,j}} = x_p^i t_q^j,\quad \frac{\partial N_x(x_p, t_q)}{\partial a_{i,j}} = i x_p^{i-1} t_q^j,$$
$$\frac{\partial N_{xx}(x_p, t_q)}{\partial a_{i,j}} = i(i-1) x_p^{i-2} t_q^j,$$

$$\frac{\partial N_t(x_p, t_q)}{\partial a_{i,j}} = j x_p^i t_q^{j-1},\quad \frac{\partial N_{tt}(x_p, t_q)}{\partial a_{i,j}} = j(j-1) x_p^i t_q^{j-2}.$$

The above computational process can similarly be employed for heat-like equation. To prevent taking much of the time on this part, the behavior of adjusting weight parameters for this equation is not provided. It should be mentioned clearly that Matlab $v7.10$ is a high quality and easy to use mathematical computing software, which researchers and students can employ to omit wasting time and enhance the accuracy of calculations.

## 3 Numerical examples

In this section two test problems with computer simulations are provided to illustrate ability and accuracy of the iterative technique proposed in this research. Furthermore, the obtained numerical results of this technique will be compared with those obtained by Taylor series method [15]. Here, the mean absolute error $E_{mid}$ *i.e.*:

$$E_{mid} = \frac{1}{(n'+1)(m'+1)} \sum_{i=0}^{n'} \sum_{j=0}^{m'} |u(x_i, t_j) - \tilde{u}(x_i, t_j)|,$$

will be implemented to compare the effectiveness of both methods.

**Example 3.1.** The following heat-like equation is considered:

$$\frac{\partial u}{\partial t} - \frac{x^2}{6} . \frac{\partial^2 u}{\partial x^2} = 0,\ 0 < x < 1,\ t > 0,$$

with initial conditions:

$$u_t(x, 0) = u(x, 0) = x^3,$$

Brought to you by | Cankaya University
Authenticated
Download Date | 12/18/19 11:48 AM

**Table 1:** Measured $E_{mid}$ errors for different number of network elements

| | ------ $n = 5$ ------ | | | ------ $n = 7$ ------ | | |
|---|---|---|---|---|---|---|
| $r$ | $n' = 5$ | $n' = 10$ | $n' = 15$ | $n' = 5$ | $n' = 10$ | $n' = 15$ |
| 1000 | $2.9323e-08$ | $2.7833e-08$ | $1.6822e-08$ | $3.5860e-11$ | $3.3621e-11$ | $2.7600e-11$ |
| 2500 | $8.7340e-09$ | $7.0041e-09$ | $6.9740e-09$ | $8.1071e-12$ | $6.0731e-12$ | $4.9617e-12$ |
| 5000 | $1.1375e-09$ | $5.6823e-10$ | $3.0116e-10$ | $3.2074e-12$ | $9.4698e-13$ | $3.6220e-13$ |
| 7500 | $7.9910e-11$ | $6.1845e-11$ | $6.4731e-11$ | $5.8193e-13$ | $2.1633e-13$ | $9.0790e-14$ |
| 10000 | $9.2155e-11$ | $5.0160e-11$ | $1.7147e-11$ | $6.0514e-14$ | $5.7311e-14$ | $4.9321e-14$ |
| | ------ $n = 9$ ------ | | | ------ $n = 11$ ------ | | |
| $r$ | $n' = 5$ | $n' = 10$ | $n' = 15$ | $n' = 5$ | $n' = 10$ | $n' = 15$ |
| 1000 | $1.6340e-12$ | $1.4333e-12$ | $1.2173e-12$ | $5.2961e-14$ | $4.8245e-14$ | $3.6175e-14$ |
| 2500 | $7.1327e-13$ | $5.0108e-13$ | $2.9388e-13$ | $2.3799e-14$ | $9.3693e-15$ | $8.6722e-15$ |
| 5000 | $2.9110e-13$ | $1.2940e-13$ | $8.4830e-14$ | $1.0861e-14$ | $6.7127e-15$ | $4.2973e-15$ |
| 7500 | $7.6931e-14$ | $6.1577e-14$ | $5.7827e-14$ | $7.1911e-15$ | $5.0409e-15$ | $2.2500e-15$ |
| 10000 | $4.2780e-14$ | $3.0091e-14$ | $1.3786e-14$ | $5.7860e-15$ | $2.0617e-15$ | $9.8476e-16$ |

**Table 2:** Absolute error comparison for different number of iterations

| $x = t$ | *TE method* | *ANNs approach* | | | |
|---|---|---|---|---|---|
| | | $r = 1000$ | $r = 2500$ | $r = 5000$ | $r = 10000$ |
| 0.1 | $1.4090e-12$ | $3.5275e-08$ | $6.5374e-09$ | $7.3595e-11$ | $8.0373e-12$ |
| 0.2 | $7.3195e-10$ | $3.2403e-08$ | $8.2796e-09$ | $3.4572e-10$ | $8.4933e-12$ |
| 0.3 | $2.8555e-08$ | $3.1375e-08$ | $8.7914e-09$ | $7.6941e-10$ | $1.0855e-11$ |
| 0.4 | $3.8598e-07$ | $1.9725e-08$ | $9.0478e-09$ | $6.7459e-10$ | $2.4573e-11$ |
| 0.5 | $2.9193e-06$ | $1.7860e-08$ | $5.4376e-09$ | $6.1150e-10$ | $3.0391e-11$ |
| 0.6 | $1.5293e-05$ | $2.3485e-08$ | $6.7328e-09$ | $9.3688e-10$ | $5.9043e-11$ |
| 0.7 | $6.2183e-05$ | $2.0389e-08$ | $6.1107e-09$ | $3.7592e-10$ | $7.6900e-11$ |
| 0.8 | $2.1005e-04$ | $1.3481e-08$ | $7.6833e-09$ | $7.8927e-11$ | $6.8627e-11$ |
| 0.9 | $6.1590e-04$ | $2.1768e-08$ | $8.3284e-09$ | $8.9380e-11$ | $2.0981e-11$ |

and the boundary conditions:

$$u(0,t) = 0,\ u(1,t) = e^t.$$

Note that the exact solution of the problem is given as $u(x,t) = x^3e^t$. Now, we intend to approximate the solution function by using the defined combination method on the domain $\Omega = [0,1]^2$. Here, we use the regular discretization technique on the given differential domain in $x$ and $t$ directions. We intend to show that the proposed four-layer feed-forward neural architecture is sufficient to solve the defined math problem. Hence, the incremental learning process begins to work by quantifying the connection weights $a_{i,j}$ (for $i = 0,\dots,n;\ j = 0,\dots,m$) with small real-valued random constants. The convergence speed of back-propagation is directly related to the learning rate and momentum constant parameters. The optimal tuning parameters for fast convergence of back-propagation gradient descent search is the inverse of the largest eigenvalue of the Hessian matrix of the defined error function, evaluated at the local point. Thus, the norm of the converged weight vector gives a good estimate of learning rate in back-propagation. In this study, for better comparison of the obtained numerical and simulation results, we had to use same quantities for rate and momentum parameters. This work makes easy to compare the obtained results from using different initial parameters and iterations numbers. In other words, using same valued learning rate and momentum constant makes better comparison of the obtained results. Then, the training patterns were used to successively adjust the connection weights until a suitable solution was found. Typically, more than one step using the training set is needed to derive an appropriate solution. To demonstrate the accuracy of technique presented in the previous section, the indicated mean absolute errors for different network parameters are shown in Table 1, for

Brought to you by | Cankaya University
Authenticated
Download Date | 12/18/19 11:48 AM

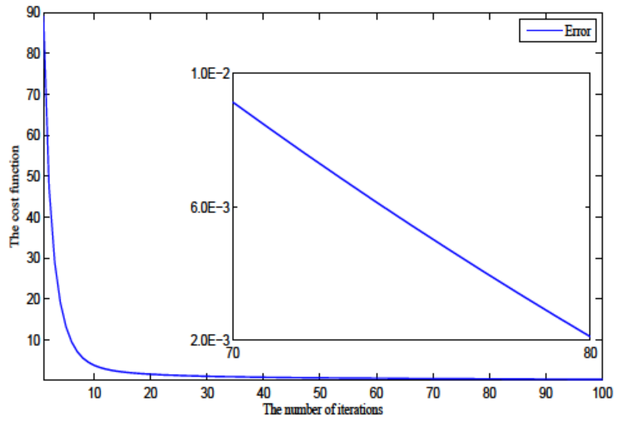

**Figure 2:** The cost function for different iteration steps

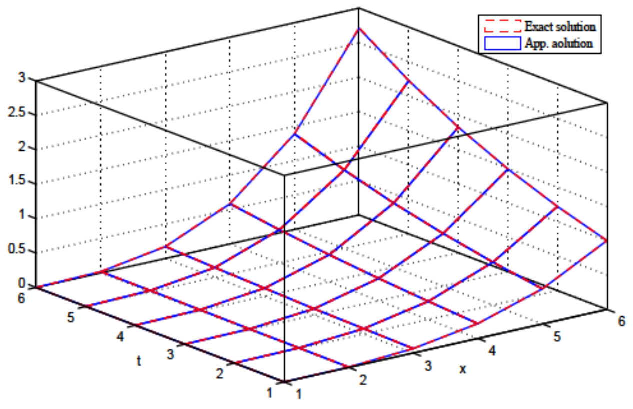

**Figure 3:** Exact and approximate solutions for $r = 1000$

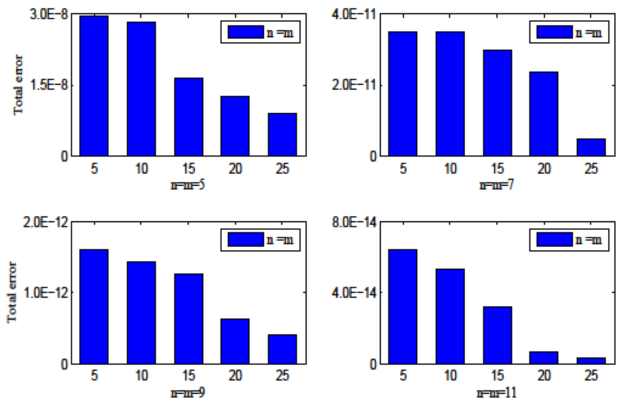

**Figure 4:** Performance of proposed neural architecture over different neural elements for $r = 1000$

$n = m$ and $n' = m'$. The absolute errors between the approximate and exact solutions on the mesh points are numerically compared for $n = m = 5$ and $n' = m' = 10$ in Table 2. As can be seen, with increasing number of iterations the ANNs approach offers more accurate approximations rather than the Taylor series method of order 5. The indicated error function is plotted in Figure 1, for $m = n = 5$ and $m' = n' = 5$. It can be easily concluded that by increasing the number of iterations, the network error is rapidly reduced until it go to zero. The exact and approximate solutions are plotted in Figure 3. The performance of proposed neural structure for different control elements is using the mean absolute error function in Figure 4.

**Example 3.2.** Consider the following one-dimensional wave-like equation:

$$\frac{\partial^2 u}{\partial t^2} - \frac{x^2}{2} \cdot \frac{\partial^2 u}{\partial x^2} = 0, \ 0 < x < 1, \ t > 0,$$

with initial conditions:

$$u(x, 0) = x + x^2, \ u_t(x, 0) = 0,$$

boundary conditions:

$$u(0, t) = 0, \ u(1, t) = 1 + cosh(t),$$

and exact solution:

$$u(x, t) = x + x^2 \cosh(t).$$

Similarly, the proposed neural architecture has been employed for approximating solution of this initial-boundary problem on square $\Omega = [0, 1] \times [0, 0.5]$ for $m = n = 5$ and $m' = n' = 10$. The obtained numerical results are presented in Table 2. We are now allowed to claim that the combination method proposed in this paper can be applied to accurately approximate the unknown functions to any desired degree of preciseness. In particular, thanks to this unique characteristic of artificial neural networks, the algorithmic power series method can be converted into an iterative non-algorithmic one. Here, if we consider an arbitrarily large number of iterations, the proposed boosting method will be able to approximate the unknown function with high precision.

## 4 Conclusion

Certain properties of artificial neural networks are typically configured to distinguish the networks based on iterative approach from other classical numerical methods. In this research, a combination of ANNs approach and power series methods, was used for numerical solution of two special types of partial differential equations. Wave equations and heat equations are considered as two main boundary-initial value partial differential equations which have played pivotal part in modeling physics phenomena. The proposed approach could convert solving

Brought to you by | Cankaya University
Authenticated
Download Date | 12/18/19 11:48 AM

**Table 3:** Absolute error comparison for $r = 10000$

| | $t = 0$ | | $t = 0.1$ | |
|---|---|---|---|---|
| $x$ | *TE method* | *ANNs approach* | *TE method* | *ANNs approach* |
| 0.1 | 0 | 0 | $1.3891e-11$ | $9.8640e-12$ |
| 0.2 | 0 | 0 | $5.5565e-11$ | $5.8647e-11$ |
| 0.3 | 0 | 0 | $1.2502e-10$ | $3.8402e-11$ |
| 0.4 | 0 | 0 | $2.2226e-10$ | $2.8223e-11$ |
| 0.5 | 0 | 0 | $3.4728e-10$ | $4.7073e-11$ |
| 0.6 | 0 | 0 | $5.0009e-10$ | $6.9314e-11$ |
| 0.7 | 0 | 0 | $6.8068e-10$ | $3.8425e-11$ |
| 0.8 | 0 | 0 | $8.8905e-10$ | $5.8014e-11$ |
| 0.9 | 0 | 0 | $1.1252e-09$ | $7.3592e-11$ |
| | $t = 0.2$ | | $t = 0.3$ | |
| $x$ | *TE method* | *ANNs approach* | *TE method* | *ANNs approach* |
| 0.1 | $8.8952e-10$ | $2.7518e-11$ | $1.0141e-08$ | $2.7468e-11$ |
| 0.2 | $3.5581e-09$ | $2.8677e-11$ | $4.0565e-08$ | $4.3099e-11$ |
| 0.3 | $8.0057e-09$ | $5.0046e-11$ | $9.1272e-08$ | $4.4029e-11$ |
| 0.4 | $1.4232e-08$ | $3.9617e-11$ | $1.6226e-07$ | $3.6418e-11$ |
| 0.5 | $2.2238e-08$ | $4.8135e-11$ | $2.5353e-07$ | $3.9214e-11$ |
| 0.6 | $3.2023e-08$ | $6.0175e-11$ | $3.6509e-07$ | $4.7109e-11$ |
| 0.7 | $4.3587e-08$ | $6.1482e-11$ | $4.9692e-07$ | $6.9243e-11$ |
| 0.8 | $5.6930e-08$ | $4.0841e-11$ | $6.4904e-07$ | $7.2390e-11$ |
| 0.9 | $7.2051e-08$ | $3.6940e-11$ | $8.2144e-07$ | $7.8814e-11$ |
| | $t = 0.4$ | | $t = 0.5$ | |
| $x$ | *TE method* | *ANNs approach* | *TE method* | *ANNs approach* |
| 0.1 | $5.7052e-08$ | $3.8425e-11$ | $2.1799e-07$ | $3.9627e-11$ |
| 0.2 | $2.2821e-07$ | $3.1208e-11$ | $8.7194e-07$ | $3.0281e-11$ |
| 0.3 | $5.1347e-07$ | $6.0172e-11$ | $1.9619e-06$ | $3.0127e-11$ |
| 0.4 | $9.1283e-07$ | $4.8466e-11$ | $3.4878e-06$ | $4.8692e-11$ |
| 0.5 | $1.4263e-06$ | $1.4370e-11$ | $5.4496e-06$ | $6.3484e-11$ |
| 0.6 | $2.0539e-06$ | $1.3692e-11$ | $7.8475e-06$ | $6.9107e-11$ |
| 0.7 | $2.7955e-06$ | $8.2410e-11$ | $1.0681e-05$ | $8.1123e-11$ |
| 0.8 | $3.6513e-06$ | $7.3596e-11$ | $1.3951e-05$ | $9.3271e-11$ |
| 0.9 | $4.6212e-06$ | $7.1104e-11$ | $1.7657e-05$ | $1.3504e-10$ |

a differential problem into related optimization minimizing problem. This work combined initial and boundary conditions of the problem, which could be easily modeled with suitable network architecture. Discretizing the differential domain and then using the back-propagation learning algorithm lead to solveing the optimization problem for the unknown series of coefficients. The validity of our method was based on the supposition that the convergence rate quickly rises by increasing the number of node points and learning steps. However, the initial values for the network parameters had a considerable impact. The learning rate and momentum constant were sensitive tools that were considered as convergence speed control parameters. Inappropriate choices for these parameters led to a lack of convergence or an excessive increase in the number of repeating steps. To make a better description of the offered technique, one numerical example was presented with computer simulations. Also, comparison of numerical results with exact solutions and those of another classical method has helped us to precisely understand this exercise. The achieved results support our claim that the designed neural architecture gives better

Brought to you by | Cankaya University
Authenticated
Download Date | 12/18/19 11:48 AM

convergent approximation without any restrictive assumptions. However, most of the equations belong to mathematical applications in real-world problems; therefore they require complex solutions. According to the numerical results obtained from two examples, it was natural to claim that the proposed procedures were valid and possessed unique properties along with high efficiency. The proposed method was more efficient than other methods. With a little care in the performance of this method, it can be easily concluded that our combination technique can be classified in row of non-algorithmic methods. Despite having control levers such as learning rate, momentum constant or variety of learning algorithms and cost functions increase the accuracy in determining the mathematical mystery. It is obvious that most classical methods are not available to solve a variety of complex mathematical problems. In many cases, these non-algorithmic methods can solve difficult mathematical problems. Research in this area can provide great benefits to the related fields, while extension of the fractional partial differential equations can be a milestone for the near future research.

## References

[1] Al-Mazmumy M., Al-Malki H., Some modifications of Adimian decomposition method for ninlinear partial differential equations, IJRRAS, 2015, 23(2), 164-173.

[2] Boukehila A., Benmostefa F.Z., Solution process of a class of differential equation using Homotopy analysis wiener hermite expansion and perturbation technique, Int. Journal of Math. Analysis, 2014, 8(4), 167-186.

[3] Cakir M., Arslan D., The Adomian decomposition method and the differential transform method for numerical solution of multi-pantograph delay differential equations, Applied Maths., 2015, 6(3), 1332-1343.

[4] Desail Kh.R., Pradhan V.H., Solution by Homotopy perturbation method of linear and nonlinear diffusion equation, Inte. Journal of Emerging Tech., 2013, 3(4), 169-175.

[5] El-Tawil M.A., Fareed A., Solution of stochastic cubic and quintic nonlinear diffusion equation using WHEP, pickard and HPM Methods, Open Journal of Disc. Math., 2011, 1(1), 6-21.

[6] Fuller R., Neural fuzzy systems, Abo Akademi University press, 2005.

[7] Graupe D., Principles of artificial neural networks (2nd Edition), World Scientific Publishing, 2007.

[8] Hanss M., Applied Fuzzy Arithmetic: An introduction with engineering applications, Springer-Verlag, Berlin, 2005.

[9] Hassoun M.H., Fundamentals of artificial neural networks, MIT Press, New york, 1995.

[10] Hendi F.A., Bakodah H.O., Al-Mazmumy M., Alzumi H., A simple program for solving nonlinear initial value problem using Adomian decomposition method, Int. J. of Rese. and Review in Appl. Scien., 2012, 12(3), 397-406.

[11] Jafarian A., Alizadeh R., A new combinative method to the two-dimensional Bratu problem, Adva. in Difference Equ., In press.

[12] Jafarian A., Kulaii L., A new combinative method to the onedimenional fractional Bratu problem,Int. J. of Dynamical Sys. and Differe. Equ., In press.

[13] Jafarian A., Mokhtarpour M., Baleanu D., Artificial neural network approach for a class of fractional ordinary differential equation, Neu. Com. and Appli., 2016, 3(2), 271-9.

[14] Jianu M., Solution of heat and wave-like equations by Adomian decomposition sumudu transform method, Rom. J. of Building, 2015, 2(1), 1-8.

[15] Jianu M., Popescu I. , Heat and wave-like equations with variable coefficients solved by Taylor series method, Rom. J. of Building, 2015, 2(1), 1-8.

[16] Jui-Sheng Ch., Chuen-Fa N., Ching-Pin L., Analytical power series solutions to the two dimensional advection-dispersion equation with distance-dependent dispersivities, Hyd. Pro., 2008, 22(4), 4670-4678.

[17] Kurulay M., Bayram M., A novel power series method for solving second order partial differential equations, Eur. J. of Pure and Appl. Math., 2009, 2(2), 268-277.

[18] Marin M., The Lagrange identity method in thermoelasticity of bodies with microstructure, Int. J. of Eng. Sci., 1994, 32(8), 1229-1240.

[19] Marin M., On existence and uniqueness in thermoelasticity of micropolar bodies, C. R. Math. Acad. Sci., 1995, 321(12), 475-480.

[20] Marin M., Marinescu C., Thermoelasticity of initially stressed bodies, asymptotic equipartition of energies, Int. J. of Eng. Sci., 1998, 36(1), 73-86.

[21] Matinfar M., Saeidy M.,Raeisi Z. , Modified variational iteration method for heat equation using He's polynomials, Bull. Math. Anal. Appl.,2011, 3(2), 238-245.

[22] Mohyud-Din S.T., Noor M.A., Noor KI., Variational iteration method for Burgers and coupled Burgers equations using He's polynomials, Zeitschrift Fur Naturforschunge Section A-A J. of Phy. Sci., 2010, 65(4), 263-267.

[23] Nuseir A., Ameina S.,Al-Hasson A., Power series solution for nonlinear system of partial differential equations, Appl. Math. Sci., 2012, 6(104), 5147-5159.

[24] Secer A., Approximate analytic solution of fractional heat-like and wave-like equations with variable coefficients using the differential transforms method, Secer Adva. in Diff. Equa., 2012, 4(3), 187-198.

[25] Tabatabaei Kh., Eluik E.C., Tabatabaei R., The differential transform method for solving heat-like and wave-like equations with variable coefficients, Turk J. Phys., 2012, 36(9), 87-98.

[26] Wazwaz A.M., The variational iteration method for solving linear and nonlinear ODEs and scientific models with variable coefficients, Cent. Eur. J. Eng. 2014, 4(1), 64-71.

Brought to you by | Cankaya University
Authenticated
Download Date | 12/18/19 11:48 AM