TJPS

TIKRIT JOURNAL OF PURE SCIENCE

Journal Homepage: http://main.tu-jo.com/ojs/index.php/TJPS/index

Solving Three Dimensions Volterra Integral Equations (TDVIE) via a Neural Network

Nahdh S. M. Al-Saif [1] , Ameen Sh. Ameen [1] , Ghaith Fadhil Abbas[2]

[1] Department of Applied Mathematics, College of Science, Anbar University, Rmady , Iraq

[2] Mathematics Department, College of Education for Pure Sciences, Tikrit University, Tikrit, Iraq

DOI: http://dx.doi.org/10.25130/tjps.23.2018.176

ARTICLE INFO.

Article history:

-Received: 20 / 12 / 2017

-Accepted: 8 / 3 / 2018

-Available online: / / 2018

Keywords: Three - dimensional Volterra Integral Equations, Artificial Neural Network, Linear Transfer Function, Levenberg-Marquardt Algorithm.

Corresponding Author:

Name: Nahdh S. M. Al-Saif

E-mail: nn_ss_m68@yahoo.com

Tel:

Abstract

The aim of this paper is present a new numerical method for solvingThree Dimensions Volterra Integral Equations using artificial neural network by design multilayer feed forward Neural Network. A multi- layers design in our proposed method consist of a hidden layer having seven hidden units and one linear output unit. Linear Transfer function used as each unit and using Levenberg-Marquardtalgorithmtraining. Moreover, examples on three- dimensional Volterra integral equations carried out to illustrate the accuracy and the efficiency of the presented method. In addition, some comparisons among proposed method and Shifted Chebyshev Polynomials method and Reduced Differential Transform Method are presented.

1. Introduction

In engineering and science problems, multidimensional integral and differential equations proved to be an important tool for modeling and solving such problems [1-2].

There are many numerical methods to solve such equations especially two dimensional integral equations, [3-12]. Three - dimensional integral equations can be solved by using some of these methods. For example,method of Degenerate Kernel Method used to solvethree-dimensionalnon- linear Volterra integral equations [13], in [14-15] differential transform method was used to solve non-linear TDVIE, and Shifted Chebyshev Polynomialsmethodare used for solving TDVIE[16].

In this study, we describe another numerical method to solve TDVIE by designing a feed forward neural network. Therefore, we consider the following TDVIE:

$$u(x,y,z) = f(x,y,z) + \int_0^z \int_0^y \int_0^x K(x,y,z,r,s,t)\, u(r,s,t)\, drdsdt \quad (1)$$

where $(x,y,z) \in D = [0,X] \times [0,Y] \times [0,Z]$, $u(x,y,z)$ is the unknown function to be found,$K(x,y,z,r,s,t)$and $f(x,y,z)$are given functions defined, respectively on D.

2. Artificial Neural Network (ANN)

An (ANN) formed from many artificial neurons(nodes equivalent to neurons of a human brain) that are joint together dependent on particular network -architecture. The goal of the neural network is to transform the inputs into significativeoutputs .

In another words (ANN) is an interconnected system of nodes by weighted arrows (equivalent to synapses between neurons). The outcome of (ANN) altered by changing of the arrow's weights. The result of the network for the data that fed to the input layer displayed by the output layer. The input nodes (represent the independent variables)that used for predicting the dependent variable (i.e. the out neurons).

In [17], (ANN) characterized by:

1- "Its pattern of connections between the neurons (called its architecture)".

2- "Methods of determining the weights on the connections (called its training or learning, algorithm)".

3- "Its activation–function".

2.1 Neural Network Structure:

The structure or topology of an artificial neuron network means the way of regulation of neuronal computational cell in the network. Particularly how the information transmitted though the network Figure 1 and how the nodes are connected. The architecture can by classified in terms of three aspects

(Number of levels or layers, Connection pattern and Information flow).

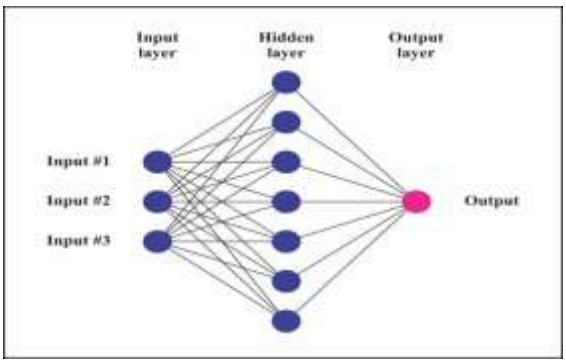

Fig. (1): Neural Network Structure

2.2 Linear Transfer Function (purelin)

The output of a linear transfer function is equal to its input:

$a = n$

as illustrated in figure 2.

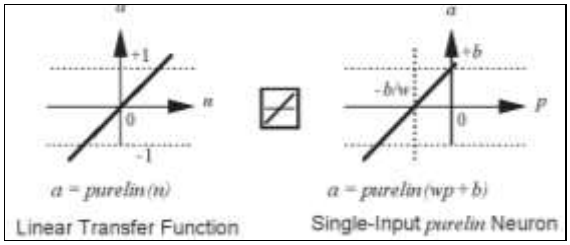

Fig. (2) Linear Transfer Function

2.3 Levenberg Marquardt Algorithm Training (trainlm)

Training neural network is basically a nonlinear squares problem, so can be solved by using a several nonlinear least squares algorithms.One of them is (LMA).We can consider (LMA) as a combination of the Gauss – Newton method and steepest descent.

For (LMA), the performance index to be optimized as

$F(w) = \sum_{p=1}^{p}[\sum_{k=1}^{k}(d_{kp} - o_{kp})^2]$ (2)

Where $w=[w_1\ w_2\ \dots w_N]^T$ consists of all weights of the network, d_{kp} is the desired value of the k^{th} output and the p^{th} pattern, o_{kp} is the actual value of the k^{th} output and the p^{th} pattern, N is the number of the weights, P is the number of pattern, and K is the number of the network output.

Equation (2) can be written as follows:

$F(w) = E^T E$ (3)

Where $E = [e_{11} \dots e_{k1} e_{12} \dots e_{k2} \dots e_{1p} \dots e_{kp}]^T$, $e_{kp}$is the trainingerror atoutput k when applying pattran p and defind as $e_{kp} = d_{kp} - o_{kp}$

$p = 1, \cdots, P$. E is the cumulative error vector (for all pattern). From equation (3) the weights are calculated using the following equation

$w_{t+1} = w_t - (J_t^T J_t + \mu_t I)^{-1} J_t^T E_t$ (4)

and the jacobian matrix is defined as

$$J = \begin{bmatrix} \frac{\partial e_{11}}{\partial w_1} & \frac{\partial e_{11}}{\partial w_2} & \dots & \frac{\partial e_{11}}{\partial w_N} \\ \frac{\partial e_{21}}{\partial w_1} & \frac{\partial e_{21}}{\partial w_2} & \dots & \frac{\partial e_{21}}{\partial w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{K1}}{\partial w_1} & \frac{\partial e_{K1}}{\partial w_2} & \dots & \frac{\partial e_{K1}}{\partial w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{1P}}{\partial w_1} & \frac{\partial e_{1P}}{\partial w_2} & \dots & \frac{\partial e_{1P}}{\partial w_N} \\ \frac{\partial e_{2P}}{\partial w_1} & \frac{\partial e_{2P}}{\partial w_2} & \dots & \frac{\partial e_{2P}}{\partial w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{KP}}{\partial w_1} & \frac{\partial e_{KP}}{\partial w_2} & \dots & \frac{\partial e_{KP}}{\partial w_N} \end{bmatrix} \quad (5)$$

where $\boldsymbol{I}$identity unit matrix, $\boldsymbol{\mu}$ the learningparameter and$\boldsymbol{J}$jacobian of m out- put error of the neural network with respect to n weights, respectively. At each iteration the μ parameter automatically adjusted in order to secure convergence, the calculation of the jacobian matrix J and the inverse of $J^T J$ square matrix of order $N \times N$ at each iteration step are the requirement of LMA.

3. Description of Method

In the current section, we will demonstrate conducing our approach to be used the approximation solution of the TDVIE.

$u(x,y,z) =$
$f(x,y,z) +$
$\int_0^z \int_0^y \int_0^x K(x,y,z,r,s,t)\, u(r,s,t)\, drdsdt$

where $(x,y,z) \in D = three\ dimension = [0,X] \times [0,Y] \times [0,Z]$, and $u(x,y,z)$is unknown function to be found.If$u_t(x,y,z,p)$ is a trial solution with adjustable parameters p, the discretized from

$Min \sum_{x_i,y_i,z_i \in D} f(x_i,y_i,z_i) +$
$\int_0^{z_i} \int_0^{y_i} \int_0^{x_i} K(x_i,y_i,z_i,r,s,t)\ u_t(r,s,t,p)\, drdsdt$

Where x ,y ,z is variables such that$(x,y,z) \in D$= three dimension=$[0,X] \times [0,Y] \times [0,Z]$.

In the our proposed approach, the trial solution $u_t$corresponds FFNN and the parameters$p$employsbiases and weights of the neural-architecture, the form for the trial-function $u_t(x,y,z)$ is $u_t(x_i,y_i,z_i,p) = G\big(x,y,z,N(x,y,z,p)\big)$

where $N(x,y,z,p)$is a single output FFNN with parameter p and n input unit feed with the input vectors x,y,z.The term G is constructed, since $u(x,y,z)$ satisfy them. This term can be formed by using a (ANN) whose biasesand weights are adjusted in order to deal with the minimization problem. The Minimized have form

$E(p) =$
$\{y_t -$
$\sum_{i=1}^{n}\big(f(x_i,y_i,z_i) +$
$\int_0^{z_i} \int_0^{y_i} \int_0^{x_i} K\big(x_i,y_i,z_i,u_t(r,s,t)\big)\, drdsdt\big)\}^2$

4.Applications to three dimensions volterra integral equation

To demonstrate the efficiency of the proposed method (ANNM), we consider the following examples and to test the accuracy of solutionsusing mean square error MSE.All programing written in the MatLab to computed the results.

Example4.1

Consider the (TDVIE)

$u(x,y,z) = f(x,y,z) - \int_0^z \int_0^y \int_0^x u(r,s,t)\, drdsdt$

where $(x,y,z) \in [0,1] \times [0,1] \times [0,1]$

And $f(x,y,z) = x + y + z + \frac{x^2yz + xy^2z + xyz^2}{2}$

has analytic function

$u(x,y,z) = x + y + z$

by applying suggested method Table (1) shows the exact, neural result, error, and men square error. Table (2) given the weight, bias, Epoch, time and performance. of the network.

Table (1): exact, neural and Accuracy of solution example (4.1)

X↓	Y	Z	Exact $u_a(x,y,z)$	Trainlm$u_t(x,y,z)$	Error = $\lvert u_t - u_a \rvert$
0.1	0.1	0.1	3.0000e-001	2.9990e-001	9.6546e-005
0.01	0.1	0.1	2.1000e-001	2.0999e-001	7.4557e-006
0.01	0.01	0.1	1.2000e-001	1.2000e-001	2.2235e-007
0.01	0.01	0.01	3.0000e-002	3.0000e-002	9.6546e-009
0.001	0.01	0.01	2.1000e-002	2.1000e-002	7.4557e-010
0.001	0.001	0.01	1.2000e-002	1.2000e-002	2.2235e-011
0.001	0.001	0.001	3.0000e-003	3.0000e-003	9.6546e-013
MSE 1.34e-009					

Table (2): weight, bias, Epoch, time and performance of the network

weight and bias					Epoch, time and performance		
Net_IW[1,1]			Net_LW[1,2]	Net_B[1]	Epoch	time	performance
0.4177	0.6665	0.8819	0.8555	0.00	9	0.00.02	1.03e-33
0.9831	0.1781	0.6692	0.6448	0.00			
0.3015	0.1280	0.1904	0.3763	0.00			
0.7011	0.9991	0.3689	0.1909	0.00			
0.6663	0.1711	0.4607	0.4283	0.00			
0.5391	0.0326	0.9816	0.4820	0.00			
0.6981	0.5612	0.1564	0.1206	0.00			

Example 4.2

Consider the (TDVIE).

$u(x,y,z) =$

$f(x,y,z) - 24x^2y \int_0^z \int_0^y \int_0^x u(r,s,t)\, drdsdt$

Where $(x,y,z) \in [0,1] \times [0,1] \times [0,1]$. And

$f(x,y,z) = x^2y + yz^2 + xyz + 24x^2y\left(\frac{xy^2z^3}{6} + \frac{x^2y^2z^2}{8} + \frac{x^3y^2z}{6}\right)$.

whichhasanalyticsolution

$u(x,y,z) = x^2y + xyz + yz^2$

by applying suggested method Table (3) shows the exact, neural result, errorand men square error . Table (4) given the weight, bias, Epoch, time and performance of the network.

Table (3): exact, neural and Accuracy of solution example (4.2)

X↓	Y	Z	Exact $u_a(x,y,z)$	Trainlm$u_t(x,y,z)$	Error = $\lvert u_t - u_a \rvert$
0.1	0.1	0.1	3.0000e-003	2.9601e-003	3.9873e-005
0.01	0.1	0.1	1.1100e-003	1.1100e-003	3.8101e-008
0.01	0.01	0.1	1.1100e-004	1.1100e-004	3.6387e-010
0.01	0.01	0.01	3.0000e-006	3.0000e-006	3.4274e-011
0.001	0.01	0.01	1.1100e-006	1.1100e-006	3.4096e-014
0.001	0.001	0.01	1.1100e-007	1.1100e-007	3.3924e-016
0.001	0.001	0.001	3.0000e-009	3.0000e-009	3.3713e-017
MSE 2.27e-010					

Table (4): weight, bias, Epoch, time and performance of the network

weight and bias					Epoch, time and performance		
Net_IW[1,1]			Net_LW[1,2]	Net_B[1]	Epoch	time	performance
0.4401	0.9577	0.2548	0.0067	0.4609	10	0.00.01	9.67e-10
0.5271	0.2407	0.2240	0.6022	0.7702			
0.4547	0.6761	0.6678	0.3868	0.3225			
0.8754	0.2891	0.8444	0.9160	0.7847			
0.5181	0.6718	0.3445	0.0012	0.4714			
0.9436	0.6951	0.7805	0.4624	0.0358			
0.6377	0.0680	0.6753	0.4243	0.1759			

Example 4.3:
Let the following (TDVIE)[15]
$u(x,y,z) = f(x,y,z) + \int_0^z \int_0^y \int_0^x u(r,s,t)\, drdsdt$
where $(x,y,z) \in [0,1] \times [0,1] \times [0,1]$.
and $f(x,y,z) = e^{x+y} + e^{x+z} + e^{y+z} - e^x - e^y - e^z + 1$.
hasanalyticsolution
$u(x,y,z) = e^{x+y+z}$
by applying suggested method Table (5) shows the exact, neural result, error,and men square error. Table (6) given the weight, bias, Epoch, time and performance of the network.

Table (5): Exact, neural and Accuracy of solution example (4.3)

X↓	Y	Z	Exact $u_a(x,y,z)$	Trainlm$u_t(x,y,z)$	Error = $\lvert u_t - u_a \rvert$
0.1	0.1	0.1	1.3499e+000	1.3507e+000	1.8419e-004
0.01	0.1	0.1	1.2337e+000	1.2338e+000	8.5500e-005
0.01	0.01	0.1	1.1275e+000	1.1275e+000	8.3164e-006
0.01	0.01	0.01	1.0305e+000	1.0305e+000	1.6259e-007
0.001	0.01	0.01	1.0212e+000	1.0212e+000	7.8907e-008
0.001	0.001	0.01	1.0121e+000	1.0121e+000	7.8634e-009
0.001	0.001	0.001	1.0030e+000	1.0030e+000	1.6224e-010
MSE 1.13 e-007					

Table (6): weight, bias, Epoch, time and performance of the network

weight and bias					Epoch, time and performance		
Net_IW[1,1]			Net_LW[1,2]	Net_B[1]	Epoch	Time	performance
0.9138	0.1704	0.4022	0.3508	0.2992	8	0.00.02	5.63e-07
0.7067	0.2578	0.6207	0.6855	0.4526			
0.5578	0.3968	0.1544	0.2941	0.4226			
0.3134	0.0740	0.3813	0.5306	0.3596			
0.1662	0.6841	0.1611	0.8324	0.5583			
0.6225	0.4024	0.7581	0.5975	0.7425			
0.9879	0.9828	0.8711	0.3353	0.4243			

To study accurate and efficient. Know comparison amongsuggest method (ANNM) with Shifted Chebyshev Polynomial method (SCPM) [16] and Reduced differential Transform method (RDTM) [14]. depended on absolute error.

Table (7): Absolute error of (SCPM), (RDTM) and (ANNM).

X↓	Y	Z	Exact u(x,y,z)	SCP Method	RDT Method$\mathbf{u_2}$	ANN method
0.1	0.1	0.1	1.3499e+000	3.3089e-002	2.0875e-004	1.8419e-004
0.01	0.1	0.1	1.2337e+000	2.1664e-002	1.9079e-004	8.5500e-005
0.01	0.01	0.1	1.1275e+000	1.1933e-002	1.7437e-004	8.3164e-006
0.01	0.01	0.01	1.0305e+000	3.6683e-003	1.7045e-007	1.6259e-007
0.001	0.01	0.01	1.0212e+000	2.5502e-003	1.6893e-007	7.8907e-008
0.001	0.001	0.01	1.0121e+000	1.4508e-003	1.6741e-007	7.8634e-009
0.001	0.001	0.001	1.0030e+000	3.6986e-004	1.6704-0010	1.6224e-010

Conclusion

Analytic solution of (TDVIE) are usually difficult, many cases required numerical solutions. In this paper, we introduced a new numerical method to solve TDVIE. The results indicate minimal mean square errorandwerecompared with the solution for some examples that solved by other methods (RDTM) and (SCPM). it was shown that the present method (ANNM) is easy implemented and highly accurate approximation solution depended on absolute error. In general the (ANNM) is powerful tool for solving (TDVIE) .

References

1. Atkinson, K.E. (1997). The Numerical Solution of Integral Equations of the Second Kind. Cambridge University Press.
2. Chari M. V. K. and Salon S. J. (2000); Numerical Methods in Electromagnetism. Academic Press.
3. Borhanifar A. and Sadri K. (2014); Numerical solution for system of two- dimensional integral equation by using Jacobi operational collocation method. Sohag J. Math. 1, **No. 1**, , 15-26.
4. Hendi F. A. and Bakodah H.O. (2011); Discrete Adomian Decomposition Solution of Non- linear Mixed Integral Equation. Journal of American Science,**7 (12),**1081- 1084
5. Mirzaee F. and Hadadiyan E. (2012),; Approximate solutions for mixed nonlinear Volterra-Fredholm type integral equations via modified block-pulse functions. Journal of the Association of Arab Universities for Basic and Applied Science12, 65- 73.

6. Ziyaee F., Tari A.(2014); Regularization Method for Two- dimensional Fredholm Integral Equations of the First Kind. International Journal of Nonlinear Science. Vol.18 **No.3**, 189- 194.
7. Saberi Nadjagi J., Reza NavidSamadi O. and Tohid E. I (2011); Numerical Solution of two-dimensional Volterra Integral Equations by Spectral Galerkin Method. Journal of Applied Mathematics and Bioinformatics, **vol. 1, no. 2,** , 159- 174.
8. Abdou M.A., Badr A.A. and Soliman M.B. (2011); On a method for solving a two- dimensional non-linear integral equation of the second kind. J. Comput. Appl. Math., 235, 35893598
9. Saeed R. K. and Berdawood K. A. (2016),; Solving Two- dimensional Linear Volterra- Fredholm Integral Equations of Second Kind by Using Successive Approximation Method and method of Successive Substitutions. ZANCO Journal of Pure and Applied Sciences. 28 (2); 35-46.
10. Sadigh Behzadi Sh. (2012); The Use of Iterative Methods to Solve Two - Dimensional Nonlinear Volterra – FredholmIntegro - Differential Equations. Communication in Numerical Analysis. 1-20.
11. AVAZZADEH Z. and HEYDARI M. (2012); Chebyshev polynomials for solving two-dimensional linear and nonlinear integral equations of the second kind. Computational and applied Mathematics. **Volume 31, No.1**, 127- 142.
12. Avazzadeha Z., Heydarib M. and Loghmania G. B. (2011); A Comparison between Solving Two-Dimensional Integral Equations by the Traditional Collocation Method and Radial Basis Function. Applied Mathematics Sciences, **Vol. 5, no. 23,** 1145-1152.
13. Basseem(2015); Degenerate Kernel Method for Three Dimension Nonlinear Integral Equations of the Second Kind. University Journal of Integral Equations 3,61-66.
14. Ziqan A., Armiti S. and I. Suwan(2016); Solving three- dimensional Volterra integral equation by the reduced differential transform method. International journal of Applied Mathematical Research. **5 (2)** 103-106.
15. Bakhshi M. and Asghari-Larimi M. (2012); Three- dimensional differential transform method for solving three-dimensional Volterra integral equations. The Journal of Mathematics and Computer Science **Vol. 4 No. 2** 246-256.
16. Mohamed D. Sh.: Shifted Chebyshev polynomials for Solving Three- Dimensional Volterra Integral Equations of the second kind. https://www.researchgate.net/publication /308692522
17. Hristev R. M. , (1998). The ANN Book.GNU public license, Edition 1 .

حل معادلة فولتيرا التكاملية ذات البعد الثلاثي

ناهض سليم محمد[1] ، امين شامان امين[1] ، غيث فاضل عباس[2]

[1]قسم الرياضيات التطبيقية ، كلية العلوم ، جامعة الانبار ، رمادي ، العراق

[2]قسم الرياضيات ، كلية التربية للعلوم الصرفة ، جامعه تكريت ، تكريت ، العراق

الملخص

في الآونة الأخيرة ازداد اهتمام الباحثون في ايجاد طرق عددية جديدة لحل معادلة فولتيره التكاملية ذات البعد الثلاثي .

ان الهدف الرئيسي لهذا البحث هو تقديم طريقة عددية جديدة لحل هذا النوع من المعادلات باستخدام الشبكات العصبية الصناعية (ANN)، حيث تم تصميم شبكة عصبية ذات تغذية امامية (FFNN) سريعة. يتالف هذا التصميم من طبقات متعددة حيث يحتوي على طبقة واحدة خفية تحتوي على سبعة وحدات خفية تستخدم الدالة الخطية (purelin) وطبقة واحدة للإخراج، تم تدريب الشبكة باستخدام خوارزمية Levenberg-Marquardt (traianlm) .

ولبيان دقة وكفاءة الطريقة المقدمة تم مقارنة نتائج الامثلة التوضيحية مع الحلول المضبوطة لهذه الأمثلة اضافة الى مقارنتها مع طريقة متعددة شبيشيف (SCPM) وطريقة التحويل التفاضلي المنخفض (RDTM) ومن خلال المقارنة تبين بان الطريقة ذات كفاءة ودقة عالية وذات خطأ صغير جدا.