Cumhuriyet Üniversitesi Fen Fakültesi
Fen Bilimleri Dergisi (CFD), Cilt:36, No: 3 Özel Sayı (2015)
ISSN: 1300-1949

Cumhuriyet University Faculty of Science
Science Journal (CSJ), Vol. 36, No: 3 Special Issue (2015)
ISSN: 1300-1949

Numerical Solution of ordinary Differential Equations Based on Semi-Taylor by Neural Network improvement

Somayeh EZADİ[1,*], Sahar ASKARI[2], Mitra JASEMI[3]

[1]*Young Researchers and Elite Club, South Kermanshah Branch, Islamic Azad University, Kermanshah, Iran.*

[2]*Department of Mathematics, Farhangian University (Shahid Sadougi Pardise) of Kermanshah Branch, Kermanshah, Iran.*

[3]*Department of Mathematics, Islamic Azad University of Kermanshah Branch, Kermanshah, Iran.*

Received: 01.02.2015; Accepted: 05.05.2015

Abstract. In this paper, a new approach is proposed in order to solve the differential equations of ordinary initial value based on the feed-forward neural network and Semi-Taylor series ordinary differential equation is first replaced by a system of ordinary differential equations. A trial Solution of this System consists of two parts. The first part, that is, Semi-Taylor series contains no adjustable parameters. And the second part includes the neural network and adjustable parameters (the weights). Using modified neural network requires that training points be selected over the open interval (a, b) without training the network in the range of the first and end points. Therefore, the calculating volume containing computational error is reduced. In fact, the training points depending on the distance [a, b] selected for training neural networks are converted into similar points in the open interval (a, b) using the new approach, then the network is trained in these similar areas.In comparison with existing similar neural networks, the proposed model provides solutions with high accuracy. Numerical examples with simulation results illustrate the effectiveness of the proposed model.

Keywords: Ordinary Differential Equations, Semi-Taylor, MLP Neural Network, bfgs Technique

1. INTODUCTION

Differential equations are used as a powerful tool in solving many problems in various fields of human knowledge, such as physics, chemistry, mechanics, economics, etc. One application of the differential equation is turning problems and natural phenomena into differential equations, then, by solving the differential equation the answer is described and the phenomena are calculated. Usually many of these problems do not have analytical solutions or their solutions may have certain implications. Many researchers have tried to approximate the solutions of these equations and proposed a lot of algorithms such as Runge-Kotta, Adomian, Adam-Beshforth, Adam-Molton, Predictor-Corrector methods and other methods. In recent years neural networks for estimation of ordinary differential equations (ODE) and partial differential equations (PDE) as well as the fuzzy differential equation (FDEs) have been used. In 1990, Lee and Kang [1] used parallel processor computers to solve a first order differential equation with Hopfield neural network models. Meade and Fernandez [2, 3] solved linear and nonlinear ordinary differential equations using feed forward neural networks architecture and B1-splines. It is not easy to extend these techniques to multidimensional domains [4]. Lagaris and colleagues (1998) used artificial neural networks to solve ordinary differential equations (ODE s) and partial differential equations (PDE) with the initial value and boundary value problems [5]. In Comparison with Jamme and Liu [6], Malek and Shekari presented numerical method based on neural network and optimization techniques which the higher-order differential equations answers approximates by finding a package form analysis of specific functions [7], In 2010, Effati and Pakdaman used artificial neural networks to solve fuzzy differential equations [8] which is a combination of two terms: The first term is related to the initial condition or boundary and the second term contains parameters related to the neural

*Corresponding author. *Email address*: *somayeh.ezadi@yahoo.com*

Special Issue: The Second National Conference on Applied Research in Science and Technology

http://dergi.cumhuriyet.edu.tr/cumuscij

network. The used neural network is a two-layer network with one hidden layer. Our previous work has improved neural network with second transfer function used to solve ordinary differential equations [9] and also Taylor-like neural network method for solving ordinary differential equations [10].

In this study, we combine and introduce methods like Taylor's, improved neural network and optimization techniques to solve ordinary differential equations. This paper is divided into five sections. In Section 2, we introduce the basic definitions and required theorems. In Section 3, the proposed model for solving the ordinary differential equations is introduced. In Section 4, illustrative example is discussed. Section 5 contains the conclusions.

2. MATERIAL and METHODS

This section provides the necessary Definitions and required theorems that are usedto propose the model.

Definition 1. (MLP Network training)

To learn the feed forward neural networks, the law of error propagation is used that is based on the error correction learning rule. Therefore, to calculate sensitivities for the different layers of neurons in the MLP network the derivation of conversion neurons functions is required. So, functions are used that have derivation. One of these functions is sigmoid or hyperbolic tangent function whose characteristics were explained in the previous section. The Error function is described in the following sections.

Theorem 1. (The World approximation Builder) The MLP network with one hidden layer with a sigmoid functions (Hyperbolic tangent function) in the middle layer and linear transformation functions in output layer are able to approximate all functions in any degree of the integral of the square (see [11]).

Definition 2. (BFGs Teqnique)

To minimize this unconstrained optimization problem, minimization techniques such as the steepest descent method and the conjugate gradient or quasi-Newton methods can be applied. The Newton method is one of the important algorithms in nonlinear optimization. The main disadvantage of the Newton method is that it is necessary to evaluate the second derivative matrix (Hessian matrix). Quasi-Newton methods were originally proposed by Davidon in 1959 and were later developed by (Fletcher and Powell, 1963). The most fundamental idea in quasi-Newton methods is the requirement to calculate an approximation of the Hessian matrix. Here, the quasi-Newton BFGS (Broyden Fletcher Goldfarb Shanno) method is used. This method is quadratic (see [12]).

3. THE NEW MODEL

Consider the following initial value first order differential equation:

$$\begin{cases} \dfrac{dy(x)}{dx} = f(x, y(x)), \quad x \in [a,b] \\ y(a) = \alpha. \end{cases} \tag{1}$$

The trial function may be written in the form of

$$y_T(x,p) = U(x) + x\mathrm{N}(x,p) \tag{2}$$

The Taylor series of a real or complex function y(x) that is infinitely differentiable in a neighborhood of a real or complex number is the power series

$$y(x) = y(x_0) + \frac{y'(x_0)(x - x_0)}{1!} + \frac{y''(x_0)(x - x_0)^2}{2!} + \frac{y'''(x_0)(x - x_0)^3}{3!} + \dots$$

that can be written in the more compact sigma notation as $y(x) = \sum_{n=0}^{\infty} \frac{y^{(n)}(x_0)}{n!}(x - x_0)^n$.

Where n! denotes the factorial of n. Exponential function $e^x = \sum_{n=0}^{\infty} \frac{y^{(n)}}{n!}$ For all x.

Praise ordinary function on basis Semi-Taylor series

$$y(x) = y(x_0) + \frac{y(x_0)(x - x_0)}{1!} + \frac{y(x_0)(x - x_0)^2}{2!} + \frac{y(x_0)(x - x_0)^3}{3!} + \dots$$

Now praise order function on basis Semi-Taylor series

$$U(x) = \left(1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}\right) y(x_0) \tag{3}$$

using (3), (2) may be rewritten as the following form:

$$y_T(x, p) = \left(1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}\right) y(x_0) + x\mathrm{N}(x, p) \tag{4}$$

Therefore

$$y'_T(x, p) = \frac{dU}{dx} + \mathrm{N}(x, p) + x\frac{\partial N}{\partial x} \tag{5}$$

where

$$\frac{dU}{dx} = \left(1 + x + \frac{x^2}{2}\right) y(x_0) \tag{6}$$

using (6), (5) may be rewritten the following form

$$y'_T(x, p) = \left(1 + x + \frac{x^2}{2}\right) y(x_0) + \mathrm{N}(x, p) + x\frac{\partial N}{\partial x} \tag{7}$$

Now consider a multilayer perceptron having one hidden layer with H sigmoid units and a linear output unit. Here we have:

$$\mathrm{N} = \sum_{i=1}^{H} V_i S(Q_i), Q = w_i(x + 1)\varepsilon + b_i \ , \ \varepsilon > 0. \tag{8}$$

Where w_i is a weight parameter from input layer to the ith hidden layer, V_i is an ith weight parameter from hidden layer to the output layer, b_i is an ith bias for the ith unit in the hidden layer, Q_i is an output of the ith hidden unit and S is an arbitrary sigmoid function.

In neural network, activation function is used for limiting the output neurons. Output's neuron is between [0, 1] or [-1.1]. In this paper, we use linear activation function and sigmoid activation function but in principle we use linear activation function and Sigmoid or (hyperbolic

tangent) activation function. The output of sigmoid function belongs to the interval [0, 1], since this function is a differentiable function. Generally, the multi-layer network with trained error back propagation algorithm is used for this function. Now consider a multilayer perceptron having one hidden layer with sigmoid units and a linear output unit.

To facilitate later discussions, we determine the first derivatives:

$$S(Q)=\frac{1}{1+\mathrm{e}^{-Q}}, \qquad S'(Q)=-S^2+S$$

The derivatives of N with respect to input x_i is

$$\frac{\partial N}{\partial x}=\sum_{i=1}^{H} V_i w_i \left[\frac{1}{1+\mathrm{e}^{-w_i(x+1)\varepsilon+b_i}}-\left(\frac{1}{1+\mathrm{e}^{-w_i(x+1)\varepsilon++b_i}}\right)^2\right] \tag{9}$$

Thus the corresponding total error function that must be minimized over all adjustable neural network parameters will be

$$EP=\sum_{i=1}^{m}\left(y'_T(x_i,p)-f\left(x_i,y_T(x_i,p)\right)\right)^2 \tag{10}$$

We can apply BFGS quasi-Newton method to minimize (10) [12].

4. NUMERICAL EXAMPLES

To show the behavior and properties of this new method, in this section, we discuss the simulation results of one example. The simulation is conducted on Matlab12, the objective function in (1) minimizer engaged is fminunc. The initial weights were randomly selected.

Example 1. Consider the following first order ODE:

$$\begin{cases}\dfrac{dy(x)}{dx}=y-x^2+1, & x\in[0,2]\\ y(0)=0.5\end{cases}$$

The exact solution of the ODE is:

$$y(x)=(x+1)^3-0.5e^x$$

The trail solution is given as

$$\begin{aligned}y_T(x)&=\left(1+\frac{x^1}{1!}+\frac{x^2}{2!}+\frac{x^3}{3!}\right)y(x_0)+xN(x,p)\\&=\left(1+\frac{x^1}{1!}+\frac{x^2}{2!}+\frac{x^3}{3!}\right)(0.5)\\&\quad+x\sum_{i=1}^{H}\frac{1}{1+e^{-w_i(x+1)\varepsilon-b_i}}.\end{aligned}$$

This error function for the H = 5 sigmoid units in the hidden layer and for m = 11 equally spaced points inside the interval [0, 2] is trained. Initial value of the weights and biases of NNM are shown in Table 1. Weights from the trained neural network are shown in Table 2. Analytical solution and trial function is shown in Fig. 1 for $0\le t\le 2$, $\varepsilon=0.5$.

Table 1. The optimal values of weights and biases.

i	1	2	3	4	5
v_i	*-0.7624*	*1.0038*	*-2.8458*	*1.5550*	*-0.6750*
w_i	*-1.2507*	*2.7548*	*3.9275*	*2.2401*	*-1.5926*
b_i	*-0.7230*	*0.0569*	*-6.0138*	*0.0401*	*7.7595*

EP(min) = 6.8078e-09

Table 2. Comparison of the exact $\mathbf{y_a}$ and approximated $\mathbf{y}_t$ solutions.

i	1	2	3	4	5	6
x_i	*0*	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*
y_t	*0.5000*	*0.6574*	*0.8293*	*1.0151*	*1.2141*	*1.4256*
y_a	*0.5000*	*0.6574*	*0.8293*	*1.0151*	*1.2141*	*1.4256*

i	7	8	9	10	11
x_t	*0.6*	*0.7*	*0.8*	*0.9*	*1*
y_t	*1.6489*	*1.8831*	*2.1272*	*2.3802*	*2.6409*
y_a	*1.6489*	*1.8831*	*2.1272*	*2.3802*	*2.6409*

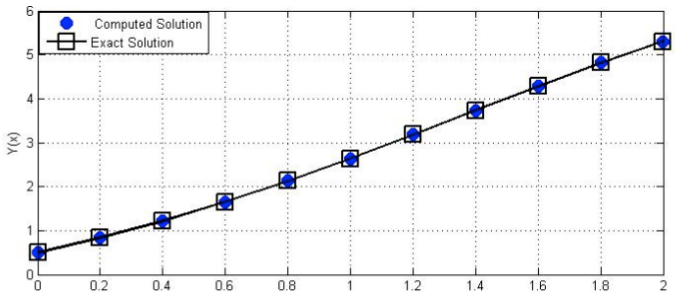

Figure 1. The exact and computed solution for Example 1.

5. CONCLUSION

In this paper, a model based on MLP neural network and semi- taylor in nonlinear optimization is proposed. We demonstrate, for the first time, the ability of neural networks and semi-taylor to approximate the solutions of ODEs. In proposed model, we use the sigmoid or hyperbolic tangant activation functions in the hidden layer of MLP neural network and obtain a solution with high accuracy. The main reason and semi- taylor for using neural networks was their applicability in function approximation.

REFERENCES

[1] Lee, H. and kang, I.S., Neural algorithms for solving differential equations, journal of computational physics 91 (1990) 110-131.
[2] Meade Jr, A.J. and Fernandez, A.A., The numerical solution of linear ordinary differential equations by feed forward neural networks, Mathematical and Computer Modelling 19 (12) (1994), 1-25.
[3] Meade Jr, A.J. and Fernandez, A.A., Solution of nonlinear ordinary differential equations by feedforward neural networks, Mathematical and Computer Modelling 20 (9) (1994) 19-44.
[4] Dissanayake, M.W.M.G. and Phan-Thien, N., Neuralnetwork based approximations for solving partial differential equations, Communications in Numerical Methods in Engineering 10 (1994) 195-201.
[5] Lagaris, I.E. and Likas, Fotiadis, A, D.I., Artificial neural networks for solving ordinary and partial differential equations, IEEE Transactions on Neural Networks 9 (5) (1998) 987-1000.
[6] Liu, Bo-An. and Jammes, B., Solving ordinary differential equations by neural networks, in: Proceeding of 13th European Simulation Multi-Conference Modelling and Simulation: A Tool for the Next Millennium, Warsaw, Poland, June, (1999) 14.
[7] Malek A., Beidokhti R. Shekar.i, Numerical solution for high order differential equations, using a hybrid neural networkOptimization method, Applied Mathematics and Computation 183, (2006) 260-271.
[8] Effati, S. and Pakdaman, M., Artificial neural network approach for solving fuzzy differential equations Information Sciences 180 (2010) 1434-1457.
[9] Ezadi, S. and Parandin, N., An application of neural networks to solve ordinary differential equations, International Journal of Mathematical Modelling & Computations 3 pp. (2013) 245- 252.
[10] Ezadi, S. and Parandin, N. and GHomashi, A., Numerical solution of fuzzy differential equations based on Semi- Taylor by using neural network, Journal of Basic and Applied Scientific Research 3(1s) pp. (2013) 477-482.
[11] Hornick K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators, Neural Networks, 2 pp. (1989) 359-366.
[12] Liu, C. and Nocedal, J., On the limited memory BFGS method for large scale optimization. Mathematical Programming, 45(3) pp. (1989) 503-528.