Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

Numerical Solution of Systems of Differential Equation with Neural Networks

Burcu Ece ALP [1], Mehmet Fatih UCAR [2]
[1] Istanbul Kultur Univesity & Turkcell Head Office, [2] Istanbul Kultur Univesity

Abstract:- We propose a physics-informed neural network (PINN) to solve boundary value system of differential equation problems. PINN is a scientific machine learning method that has been used very frequently lately to find numerical solutions of partial differential equations and offers positive results. PINNs have shown effective performance in solving a variety of differential equations, including complex derivatives and multidimensional equations. Well-trained PINNs most closely predict the numerical solutions of boundary value problems. Numerical experiments include various types of linear differential equation systems.

Keywords*:* Boundary value problems, System of differential equations, Artificial neural network, Physics-informed neural networks.

1. Introduction

Neural networks have been studied successfully in many fields and have been used to approximate the solutions of differential equations; see, e.g., [1-4]. In this research, many (unsupervised) neural networks were successfully developed without data sets; e.g. physics-informed neural networks (PINNs), where the loss function is defined by using a certain residual from the differential equation under consideration [5-6]. Additionally, PINNs use sequence points in the space-time domain as the training data set and PINNs provide suitable conditions for the solution of time-dependent, multidimensional equations [7-10]. There are some advantages of using PINNs in numerical solutions of differential equations compared to many other numerical methods. First of all, they are unsupervised learning processes, and therefore the learning process can begin without knowing the exact solution of a model differential. The exact solution is usually used to compare the solution with PINN, that is, to find the error. The most advantageous aspect of using PINNs (over traditional numerical methods) is that they apply to many different types of differential equations because the underlying equation used is (like exact solution) is used only when calculating the error. Numerical experiments show that the present PINN method provides accurate solutions on the considered computational space-time domain. In this article, we approximate the solutions of various 1D boundary value problems. Here we propose a physics informed neural network (PINN) method to solve differential equation systems with variable coefficients on a finite domain. The PINN generate approximate solutions to the boundary value problems by training to minimize the physical loss function consisting of residual [10]. Numerical experiments show that the present PINN method provides accurate solutions on the considered computational space-time domain.

2. Boundary Value Problems

Systems of ordinary differential equations have been applied to many problems in physics, engineering, biology and so on. There are many publications dealing with the linear system of second-order boundary value problems. They introduced various numerical methods. For instance, a finite difference method has been proposed in recent works [13]. We consider the following linear system of second-order boundary value problems [12]:

$$u'' + a_1u' + a_2u + a_3v'' + a_4v' + a_5v = f_1(x),$$

$$v'' + b_1v' + b_2v + b_3u'' + b_4u' + b_5u = f_2(x)$$

where $a_i(x), b_i(x), f_1(x)$ and $f_2(x)$ are given functions, and $a_i(x), b_i(x)$ are continuous,$i = 1,2,3,4,5$.

3. Physics Informed Neural Network (PINN)

Physics-informed neural networks (PINNs) are a deep learning technique used in solving differential equations [11].

It allows computers to compute the partial derivative of a value function accurately and quickly.

For time-dependent problems, we consider time t as a special component of x, and

Ω contains the temporal domain.

We use PINN method for the solution of a linear system of second order boundary value problems with the assumption that the solutions are unique.

PINNs generate approximate solutions to systems by training a neural network to minimize a loss function consisting of terms representing the misfit of the boundary conditions along the boundary of the space-time domain as well as the ODEs residual at selected point in the interior [16].

By considering$z = [u, v]^T$ PINN method assumes the following approximate function

$w_\theta(z) \coloneqq W^L\sigma^L\big(W^{L-1}\sigma^{L-1}\big(\dots\sigma^1(W^0z + b^0)\big) + b^{L-1}\big) + b^L$

θ onsists weight matrices (W) and bias vectors (b)

Weight matrices;

$W^0 \in R^{m\times 2}, W^i \in R^{m\times m} \quad (i = 1,2, \dots, L-1)$And

$W^L \in R^{1\times m}$

m denotes the number of neurons, L is the number of layers

Bias vectors;

$b^i \in R^{m\times 1}(i = 1,2, \dots, L-1)$ and $b^L \in R$.

σ (.) function is the activation function that transforms the neural network into a nonlinear form.

The approximation in (2) is an approximation function of the multilayer feed forward neural network. Here m is the number of neurons in the layers and L is the number of layers.

Equation (2) contains vectorized parameters

$\theta = W^0, b^0, W^1, b^1, \dots, W^L, b^l$

and it is necessary to optimize this approximation function to provide the given ODEs.

The residue we will use in this process, PINN approximation for eq. (2)

$R_\theta(u, v) \coloneqq \theta_t w_\theta(u, v) - L[w_\theta](u, v)$

is continuous residual function. The L operator in this function is defined as,$L: V \to R; V = w(u, v): R^2 \to R$

The PINN method minimizes the residual function on the collocation points defined on the boundary value problem calculation region, while the function tries to optimize the θ parameter. Let's take the collocation points as.

$X_i^r = \{(u_i^r, v_i^r)\}_{i=1}^{N_r} \subset [u_L, u_R].[0, T].$

Although these collocation points are random in the classical PINN method, we take them as equally spaced points. Since the derivative approximation of the equation in (5) requires x_i points with step size h .

The PINN method is obtained by minimizing the following loss functional for the solution of equation (6).

$\phi_\theta \coloneqq \phi_\theta^r(X^r) + \phi_\theta^b X_b^r$

There are three different loss functions and they belong to the residue, boundary conditions, respectively. We chose to optimize all loss functions using Adam method. Residual loss function can be written as$\phi_\theta^r(X^r) = \frac{1}{N_r}\sum_{i=1}^{N_r}|r_\theta(u_i^r, v_i^r)|^2$.

On the other hand, functions of boundary conditions are defined as

$$\phi_\theta^b X_b^r = \frac{1}{N_b}\sum_{i=1}^{N_b}\left|w_\theta(u_i^b, 0) - f(u_i^b)\right|^2.$$

Thus, the solution of the following optimization problem$\Theta^* = A = \arg\min_{\phi} \theta(x^R)$

Residue; returns to the optimum ϕ^* parameters that minimize the initial and boundary functions.

In this study, the codes are written on the Python library Tensor Flow and the optimal ϕ^* parameters were obtained with the Adam algorithm.

4. Numerical ıllustratıons

Figures and Tables

Let us assume the following differential equation systems.

$u''(x) + xu(x) + xv(x) = f_1(x)$

$v''(x) + 2xv(x) + 2xu(x) = f_2(x)$

with boundary condition can be taken from the exact solution

$u(0) = u(1) = 0, v(0) = v(1) = 0$

where $0 < x < 1$, $f_1(x) = 2$ and $f_2(x) = -2$

The exact solution of this problem for

$u(x) = x^2 - v(x) = x - x$

Numerical results are listed Table1:

Table 1. Literature studies and comparison of PINN results in different parameters.

$Epoch$	$L = 1, N = 15$
5000	$6.034541e^{-}05$
10000	$2.86675e - 07$
20000	$8.25724e - 07$

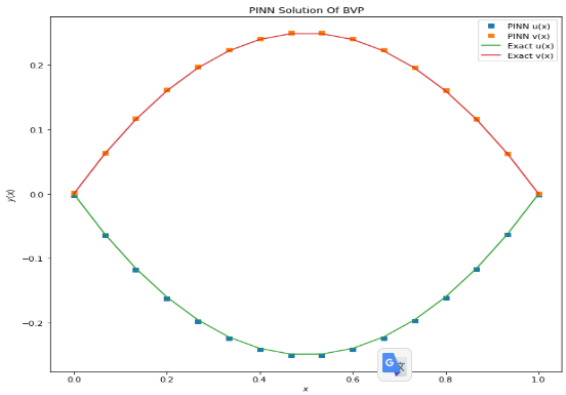

Figure 1. Comparison of PINN and analytical soulution

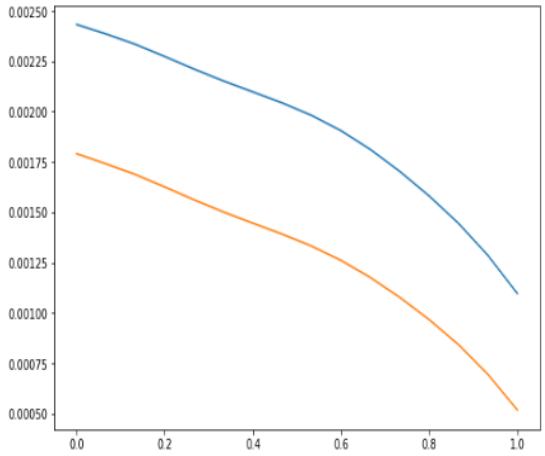

Figure 2. The absolute error surface given by the PINN model

5. Conclusion

In this article, approximate solutions and error analysis of the boundary value problems are obtained using the feed forward PINN model. Compared to the traditional numerical methods, PINNs employ automatic differentiation to handle differential operators.

Unlike numerical differentiation, automatic differentiation does not differentiate the data.

The neural networks modelled for the solutions of the problems were obtained by using the Adam algorithm in the Tensor Flow library.

With the help of numerical experiments, it has been shown that the PINN method can be an alternative convergent approach method.

The effect of the number of the collocation points, and the number of neurons in each layer on the absolute error are presented with quantitative observations.

The PINN method has been compared with the numerical methods available in the literature.

References

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

[2] L. Ruthotto, E. Haber, “Deep Neural Networks Motivated by Partial Differential Equations,” Journal of Mathematical Imaging and Vision 62:352–364 (2020).

[3] D. R. Parisi, M. C. Mariani, M. A. Laborde, “Solving differential equations with unsupervised neural networks,” Chemical Engineering and Processing 42 715-721 (2003).

[4] H. Lee, I.S. Kang, “Neural algorithms for solving differential equations,” Journal of Computational Physics, 91(1) 110-131 (1990).

[5] Z. Chen, D. Xiu, “On generalized residual network for deep learning of unknown dynamical systems,” Journal of Computational Physics, 438 (2021).

[6] Z. Chen, V. Churchill, K. Wu, D. Xiu, “Deep neural network modeling of unknown partial differential equations in nodal space,” Journal of Computational Physics 449 (2022).

[7] E. Kharazmi, Z. Zhang, G. E. Karniadakis, “hp-VPINNs: Variational physics-informed neural networks with domain decomposition”, Computer Methods in Applied Mechanics and Engineering Volume 374 113547 (2021).

[8] L. Lu, X. Meng, Z. Mao, G. E. Karniadakis, Deepxde : “A deep learning library for solving differential equations,” SIAM Review, 63 (1) 208–228 (2021).

[9] K. Shukla, P. C. D. Leoni, J. Blackshire, D. Sparkman, G. E. Karniadakis, "Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks", Journal of Nondestructive Evaluation, 39 (3) 1–20 (2020.

[10] L. Yang, X. Meng, G. E. Karniadakis, "B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data", Journal of Computational Physics,425 (2021).

[11] M. Raissi, P. Perdikaris, G.E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," Journal of Computational Physics 378 686–707 (2019).

[12] N. Caglar, H. Caglar, "B-spline method for solving linear system of second-order boundary value problems," Computers and Mathematics with Applications 57 757-762 (2009).

[13] D. Roten, T. Y. Yeh, K. B. Olsen, S. M. Day, Y. Cui "Implementation of Iwan-Type Nonlinear Rheology in a 3D High-Order Staggered-Grid Finite-Difference Method," Bulletin of the Seismological Society of America 113 (6): 2275–2291 (2023).

[14] L. Wu, H. Zhang, X. Yang, F. Wang, "A second-order finite difference method for the multi-term fourth-order integral–differential equations on graded meshes," Computational and Applied Mathematic 41: 313 (2022).