Numerical Solution for One Nonlinear Integro-Differential Equation Applying Deep Neural Network

Zurab Kiguradze

Ilia Vekua Institute of Applied Mathematics of Ivane Javakhishvili Tbilisi State University
Tbilisi, Georgia
E-mail: zurab.kiguradze@tsu.ge

Studying the propagation of an electromagnetic field into a substance, along with its mathematical modeling, investigation, and numerical solution, stands as one of the important tasks in applied mathematics. Typically, this phenomenon is associated with the generation of thermal energy, leading to alterations in the permeability of the medium and influencing the diffusion process. The mathematical representation of this process, like numerous other applied problems, results in the nonlinear partial differential and integro-differential equations and systems thereof. In a quasistationary scenario, the corresponding system of Maxwell equations takes the form outlined in [9]:

$$\frac{\partial H}{\partial t} = -\nabla \times (\nu_m \nabla \times H), \tag{1}$$

$$c_\nu \frac{\partial \theta}{\partial t} = \nu_m (\nabla \times H)^2, \tag{2}$$

where $H = (H_1, H_2, H_3)$ is a vector of the magnetic field, θ is temperature, c_ν and ν_m characterize the heat capacity and electrical conductivity of the medium which are functions of θ. As demonstrated in [3], the system represented by equations (1) and (2) can be rewritten into the following nonlinear parabolic-type integro-differential equation

$$\frac{\partial H}{\partial t} = -\nabla \times \left[a\left(\int_0^t |\nabla \times H|^2 \, d\tau \right) \nabla \times H \right], \tag{3}$$

where the function $a = a(S)$ is defined for $S \in [0, \infty)$.

Assuming that the magnetic field has the form $H = (0, 0, U)$, and $U = U(x, t)$, we get the following nonlinear integro-differential equation:

$$\frac{\partial U}{\partial t} - \frac{\partial}{\partial x} \left[a\left(\int_0^t \left(\frac{\partial U}{\partial x} \right)^2 d\tau \right) \frac{\partial U}{\partial x} \right] = 0. \tag{4}$$

The aim of the current note is to extend the investigation initiated in [8] and employ a Deep Neural Network (DNN) for the nonlinear equation (4) featuring the diffusion coefficient $a(S) = (1 + S)^p$, $0 < p \le 1$.

Thus, our goal is to apply DNN for the approximate solution of the following nonlinear initial-boundary value problem

$$\begin{gathered} \frac{\partial U(x,t)}{\partial t} - \frac{\partial}{\partial x} \left[\left(1 + \int_0^t \left(\frac{\partial U(x,t)}{\partial x} \right)^2 d\tau \right)^p \frac{\partial U(x,t)}{\partial x} \right] = f(x,t), \quad (x,t) \in \Omega, \\ U(0,t) = U(1,t) = 0, \quad t \in [0, T], \\ U(x,0) = U_0(x), \quad x \in [0, 1], \end{gathered} \tag{5}$$

where $\Omega = (0,\,1) \times (0,\,T)$, $T = const > 0$, f and U_0 are the given functions.

Qualitative and quantitative properties, as well as the numerical solution for the problem (5) and its even more intricate nonlinear counterparts, have been extensively explored in the literature (refer to, for instance, [1, 3–12, 14] and the references therein). As previously stated, our objective is to investigate an alternative approach to solving partial differential equations (PDEs) through Machine Learning methods. Specifically, we aim to train the DNN to serve as a surrogate model capable of predicting the solution of the PDE at any given point $(x,\,t) \in \Omega$. DNNs can consist of multiple layers, including input and output layers, and may feature any number of inner layers referred to as hidden layers (see, for example, Fig. 1). The deep of the network is determined by the number of hidden layers (columns of yellow circles – neurons).

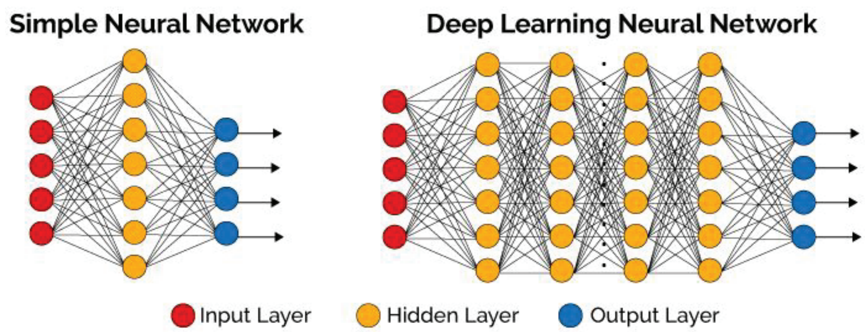

Figure 1. Example of the Neural Network architectures.

The DNN constructs approximation for the solution of problem (5) $u(x,t,\rho) \approx U(x,t)$, where $u(x,t,\rho)$ represents the function obtained from the DNN, and ρ is the variable encompassing all DNN parameters that need for optimization during the training process. As highlighted in [8],the training of the DNN necessitates a substantial amount of training data, serving as the DNN's input. Nevertheless, utilizing the DNN for approximating solutions to PDEs offers an advantage by incorporating physics, thus reducing the size of the required training data (see, for example, [2,13]).

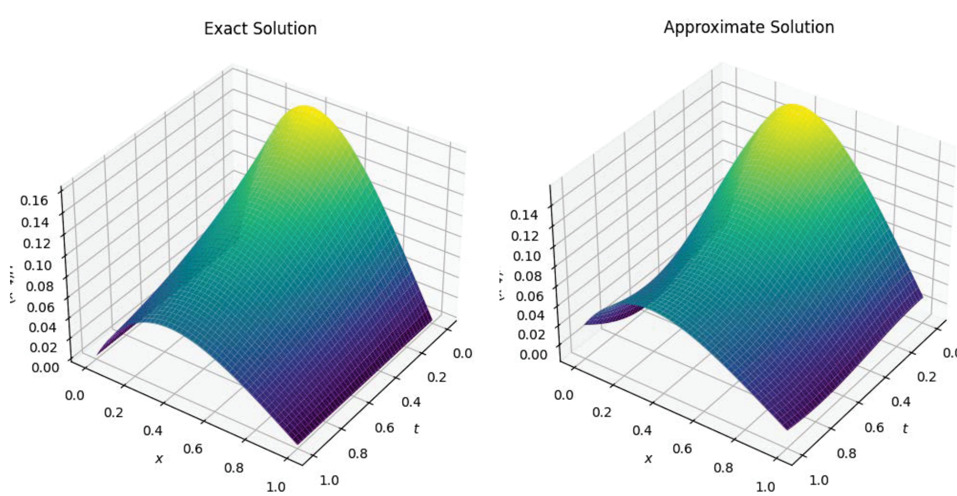

Figure 2. Exact and numerical solutions ($p = 0$).

Following the methodology outlined in [2, 8, 13], we can construct the residual of the nonlinear

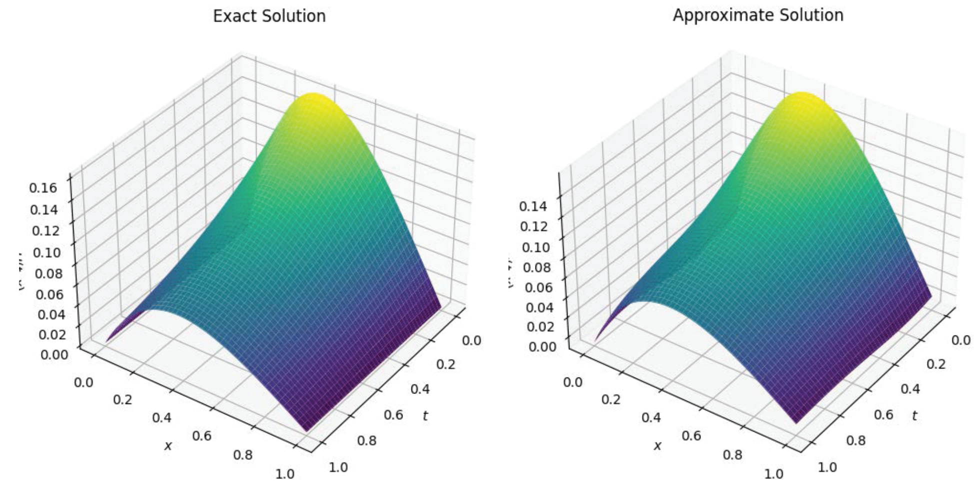

Figure 3. Exact and numerical solutions ($p = 0.5$).

problem (5) to be assessed at a designated set of training points

$$R(x,t,\rho) = \frac{\partial u(x,t,\rho)}{\partial t} - \frac{\partial}{\partial x}\left[\left(1 + \int\limits_0^t \left(\frac{\partial u(x,t,\rho)}{\partial x}\right)^2 d\tau\right)^p \frac{\partial u(x,t,\rho)}{\partial x}\right] - f(x,t). \tag{6}$$

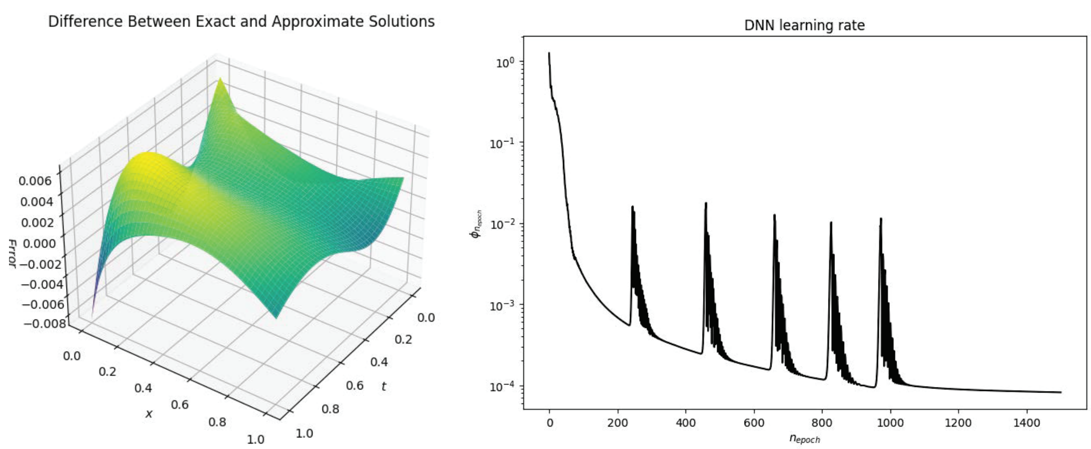

Figure 4. Difference between exact and numerical solutions and learning rate ($p = 0.5$).

Additionally, a cost function $\mathcal{F}(x,t,\rho)$ encompassing the residual (6), along with initial and boundary conditions can be built and minimized by a Deep Neural Network during the training.

In the test experiments, we adopted the same example and parameters as provided in [8]. The right-hand side $f(x,t)$ of the problem (5) was selected to yield an exact solution as follows $U(x,t) = x(1-x)\exp(-x-t)$, accompanied by the corresponding initial solution $U_0(x) = x(1-x)\exp(-x)$. The training of the neural network was conducted using the NumPy library for scientific computing and the TensorFlow library for machine learning.

In Fig. 2 we replicated results of the numerical experiment given in [8] that is for the case $p = 0$ in the problem (5).

The results of the numerical experiment for $p = 0.5$ is given on Fig. 3. The difference between exact and numerical solutions is given in Fig. 4 (left). In the same figure, the DNN learning rate for 1500 epochs is given on the right.

Acknowledgments

This work is supported by the Shota Rustaveli National Science Foundation (SRNSFG), # FR-21-2101.

References

[1] Yu. Bai and P. Zhang, On a class of Volterra nonlinear equations of parabolic type. *Appl. Math. Comput.* **216** (2010), no. 1, 236–240.

[2] J. Blechschmidt and O. G. Ernst, Three ways to solve partial differential equations with neural network-a review. *GAMM-Mitt.* **44** (2021), no. 2, Paper no. e202100006, 29 pp.

[3] D. G. Gordeziani, T. A. Dzhangveladze and T. K. Korshiya, Existence and uniqueness of the solution of a class of nonlinear parabolic problems. (Russian) *Differentsial'nye Uravneniya* **19** (1983), no. 7, 1197–1207; translation in *Differ. Equ.* **19** (1984), 887–895.

[4] T. Jangveladze, On one class of nonlinear integro-differential parabolic equations. *Semin. I. Vekua Inst. Appl. Math. Rep.* **23** (1997), 51–87.

[5] T. Jangveladze, Investigation and numerical solution of nonlinear partial differential and integro-differential models based on system of Maxwell equations. *Mem. Differ. Equ. Math. Phys.* **76** (2019), 1–118.

[6] T. Jangveladze and Z. Kiguradze, Large time behavior of solutions to one nonlinear integro-differential equation. *Georgian Math. J.* **15** (2008), no. 3, 531–539.

[7] T. Jangveladze, Z. Kiguradze and B. Neta, *Numerical Solutions of Three Classes of Nonlinear Parabolic Integro-Differential Equations.* Elsevier/Academic Press, Amsterdam, 2016.

[8] Z. Kiguradze, Approximate solution for heat equation applying neural network. *Reports of QUALITDE* **1** (2022), 135–138;
`https://rmi.tsu.ge/eng/QUALITDE-2022/Kiguradze_Z_workshop_2022.pdf`.

[9] L. D. Landau and E. M. Lifshitz, *Electrodynamics of Continuous Media.* (Russian) Gosudarstv. Izdat. Tehn.-Teor. Lit., Moscow, 1957.

[10] G. I. Laptev, Quasilinear parabolic equations that have a Volterra operator in the coefficients. (Russian) *Mat. Sb. (N.S.)* **136(178)** (1988), no. 4, 530–545; translation in *Math. USSR-Sb.* **64** (1989), no. 2, 527–542.

[11] Y. P. Lin and H.-M. Yin, Nonlinear parabolic equations with nonlinear functionals. *J. Math. Anal. Appl.* **168** (1992), no. 1, 28–41.

[12] N. T. Long and P. N. D. Alain, Nonlinear parabolic problem associated with the penetration of a magnetic field into a substance. *Math. Methods Appl. Sci.* **16** (1993), no. 4, 281–295.

[13] M. Raissi, P. Perdikaris and G. E. Karniadakis, Physics informed deep learning (Part I): data-driven solutions of nonlinear partial differential equations. *arXiv 1711.10561*, 2017; https://arxiv.org/abs/1711.10561.

[14] N. Sharma, M. Khebchareon, K. Sharma and A. K. Pani, Finite element Galerkin approximations to a class of nonlinear and nonlocal parabolic problems. *Numer. Methods Partial Differential Equations* **32** (2016), no. 4, 1232–1264.