Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 10, n. 1, 2023.

Trabalho apresentado no XLII CNMAC, Universidade Federal de Mato Grosso do Sul - Bonito - MS, 2023

**Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**

# PINNs Based on the Burgers Equation

Vitória Biesek[1] Anderson B. Modena[2] Pedro H.A. Konzen[3]
IME, UFRGS, RS

A Physics-informed neural network (PINN) is a deep learning framework for solving partial differential equations (PDEs). Deep learning is a field of machine learning by multiple levels of composition [1]. Introduced in the paper [2], the PINNs have since gain attention by its simplicity and potential efficiency as a general purpose solver for PDEs (see, for instance, [3]). In this work, we discuss on the application of PINNs to solve the Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in \mathcal{D}, t > 0, \tag{1}$$

where $\mathcal{D} = (a, b) \subset \mathbb{R}$, $\nu > 0$. A given initial and a Dirichlet boundary conditions are assumed to close the problem of solving the equation.

Burgers equation is a benchmark problem to test new numerical approaches for solving convective-diffusive PDEs. Since it has been introduced, it has been applied to the understanding of turbulent fluids, shock flows, wave propagation in combustion chambers, vehicular traffic movement, acoustic transmission and many other applications [4]. For small values of $\nu$, the convection term dominates and standard numerical discretization schemes (e.g. Finite Discrete Method, Finite Volume Method and Finite Element Method) are numerically unstable. The study of the application of PINNs to solve the Burgers equation for different values of the diffusion coefficient $\nu$ is one of the specific goals of this work.

We assume an Artificial Neural Network (ANN) of the type Multi-layer Perceptron (MLP, [5]). It has $(x, t) \in \overline{\mathcal{D}}$ as inputs and the estimate $\tilde{u} \approx u(x, t)$ as the output. It is denoted as

$$\tilde{u}(x, t) = \mathcal{N}\left(x, t; \left\{\left(W^{(l)}, \boldsymbol{b}^{(l)}, f^{(l)}\right)\right\}_{l=1}^{n_l}\right), \tag{2}$$

where $\left(W^{(l)}, \boldsymbol{b}^{(l)}, f^{(l)}\right)$ is the triple of weights $W^{(l)}$, bias $\boldsymbol{b}^{(l)}$ and activation function $f^{(l)}$ in the $l$-th layer of the network, $l = 1, 2, \ldots, n_l$. Following the PINNs approach, the training of such an ANN to solve (1) is performed by solving the following minimization problem

$$\min_{\left\{\left(W^{(l)}, \boldsymbol{b}^{(l)}, f^{(l)}\right)\right\}_{l=1}^{n_l}} \left\{\frac{1}{N_r}\sum_{i=1}^{N_r} r^2(x_{r,i}, t_{r,i}) + \frac{p}{N_b}\sum_{i=1}^{N_b}\left[\tilde{u}(x_{b,i}, t_{b,i}) - u(x_{b,i}, t_{b,i})\right]^2\right\}, \tag{3}$$

where $\{(x_{r,i}, t_{r,i})\}_{i=1}^{N_r}$ are $N_r$ selected collocation points $x, t \in (a, b) \times (0, T]$ and $\{(x_{b,i}, t_{b,i})\}_{i=1}^{N_b}$ are $N_b$ selected initial and boundary points. It is usually necessary to consider a penalty parameter $p$ to increase the importance of the boundary points on the computation of the loss function, the functional to be minimized. Moreover, $r(x, t)$ denote the PDE residual, i.e.

$$r(x, t) := \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in \mathcal{D}, t > 0. \tag{4}$$

[1] vitoriabiesek@gmail.com
[2] andersonmodena@hotmail.com
[3] pedro.konzen@ufrgs.br

 

The residual requires the computation of the derivatives of $u$, both on time $t$ and space $x$. This can be performed by finite difference formulae, but they can be also computed directly from the ANN (2). The comparison from these both approaches is also our goal in the context of solving (1).

This work is in progress and is intended to contribute on the better understanding of the advantages and disadvantages of the applications of PINNs to solve convective-diffusive partial differential equations.

## References

[1] I. Goodfellow, Y. Bengio, and A. Courville. **Deep Learning**. London: Massachusetts Institute of Technology, 2014. ISBN: 9780262035613.

[2] M. Raissi, P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: **Journal of Computational Physics** 378 (2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045.

[3] F.F. Mata, A. Gijón, M. Molina-Solana, and J. Gómez-Romero. "Physics-informed neural networks for data-driven simulation: Advantages, limitations, and opportunities". In: **Physica A: Statistical Mechanics and its Applications** 610 (2023), p. 128415. ISSN: 0378-4371. DOI: 10.1016/j.physa.2022.128415.

[4] P.H.A. Konzen, F.S. Azevedo, E. Sauter, and P.R.A. Zingano. "Numerical Simulations with the Galerkin Least Squares Finite Element Method for the Burgers' Equation on the Real Line". In: **Tendências em Matemática Aplicada e Computacional** 18.2 (2017), pp. 287–304. DOI: 10.5540/tema.2017.018.02.0287.

[5] S. Haykin. **Neural Networks: A Comprehensive Foundation**. Delhi: Pearson, 2005. ISBN: 9788177588521.