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

Solution of advection-diffusion-reaction inverse problems with Physics-Informed Neural Networks

Roberto Mamud[1]
UFRJ, Macaé, RJ
Carlos T. P. Zanini[2]
UFRJ, Rio de Janeiro, RJ
Helio S. Migon[3]
UFRJ, Rio de Janeiro, RJ
IPRJ, Nova Friburgo, RJ
Antônio J. Silva Neto[4]
IPRJ, Nova Friburgo, RJ

Abstract. In this work, two inverse problems related to pollutant dispersion in a river considering the advection-dispersion-reaction equation are studied along with a Neural Network approach. The first inverse problem concerns the estimation of the reaction parameter in an homogeneous equation, and the second one concerns the estimation of source pollution location in the non-homogeneous case. Both inverse problems are solved by two multiplayer perceptron networks: the usual Artificial Neural Network (ANN) and the Physics-Informed Neural Network (PINN), which is a special type of neural network that includes the physical laws that describes the phenomena in its formulation . Numerical experiments related to both inverse problems with ANN and with PINN are presented, demonstrating the feasibility of the proposed approach.

Keywords. Inverse Source Problem, Parameter Estimation, Physics-Informed Neural Network.

1 Introduction

Over the last centuries, with the development of industrial production, starting with the industrial revolution, the world was capable of producing in larger quantities, improving the quality process, and reducing the time of goods production. However, with this increasing industrial process, pollution has also increased, generating, in many cases, contaminants in the air, soil, and water. Unfortunately, the leakage of pollutants into bodies of water like rivers, lakes, or oceans is a common occurrence nowadays. Here we consider the model posed by the advection-dispersion-reaction equation, which allows the computation of the concentration of a pollutant in a river.

In this context, the area of inverse problems emerges as a tool in the study of these leakages which is instrumental in the prevention and/or design and planning of remediation measures to reduce their impact. For example, when it is observed that there is a pollutant in some region of the river, it is important to understand two problems, the one of study how this concentration is evolving along the river and also discover the starting location of the leakage. In the Inverse Problem framework, the first problem translates to estimating the reaction parameter of

[1]rmamud@macae.ufrj.br
[2]carloszanini@dme.ufrj.br
[3]migon@im.ufrj.br
[4]ajsneto@iprj.uerj.br

 © 2023 SBMAC

the advection-dispersion-reaction equation based on synthetic concentration measurements from sensors. In the second problem, based on concentration measurements, we are interested in determining the location of the pollution source, [4] and [2]. These kind of inverse problems are posed in contrast with the so-called direct problems [7]. In the examples above, the direct problems, given the parameters of the medium (in the first problem) and also the source location (in the second problem), concerns estimating the pollution concentration in any region of the river at any time. In this work, we study two inverse problems in the context of pollutant dispersion in a river: the parameter and the source location estimation.

In both cases, a new kind of Neural Network is employed to solve the inverse problems, namely the Physics-Informed Neural Network (PINN). The PINN is a type of neural network introduced in a recent work [8] that takes into account the physical laws that describe the phenomena involved, along with the process of training to solve some learning tasks. The PINN modifies the mean squared error loss by adding a term representing the physical laws of the model through linear or non-linear Partial Differential Equations (PDE's), where the derivatives with respect to parameters of the network are computed using Automatic Differentiation [3]. In other words, the PINN can be training via gradient descent methods by applying the chain rule to differentiate compositions of functions using the automatic differentiation technique to compute the gradient at each iteration.

In Section 2, we consider the formulation of direct and inverse problems for the pollutant concentration computation for both problems: the reaction coefficient determination and the source determination, respectively. Section 3 is devoted to the presentation of main ideas of the PINN and also the formulation of the PINN for both inverse problems. In Section 4, we present numerical experiments related to the parameter estimation of the reaction term in the advection-dispersion-reaction equation, as well as the source location estimation of the source term. Conclusions are described in Section 5.

2 The Pollutant Dispersion Direct and Inverse Problems

In this section, we establish the direct and inverse problem formulations for estimation of the reaction coefficient and the pollution source location, respectively.

Consider the following one-dimensional transient problem:

$$\begin{cases} \partial_t c - D\partial_{xx} c + V\partial_x c + Rc = F(x,t), \text{ for } x \in (0,L), t \in (0,T) \\ c(x,0) = g_0(x), \text{ for } x \in [0,L] \\ c(0,t) = g_l, \text{ for } t \in (0,T] \\ c(L,t) = g_r, \text{ for } t \in (0,T], \end{cases} \tag{1}$$

where ∂_t, ∂_x and ∂_{xx} represent the partial derivatives of concentration c $[g/m^3]$ with respect to time, $t\,[s]$, and space $x\,[m]$, respectively. Beside this, $L\,[m]$ is the length of the domain, $D\,[m^2/s]$ is the dispersion coefficient, $V\,[m/s]$ is the velocity, $R\,[1/s]$ is the reaction coefficient, F is the source term, g_0 is the initial concentration and g_l and g_r are the known concentration values at the boundary, in which the compatibility conditions $g_l(0) = g_0(0)$ and $g_r(0) = g_0(L)$ hold. This problem has a unique solution, provided that u, g_0, g_l andg_r belong to appropriated function spaces, see for example [5].

2.1 The Reaction Parameter Estimation Problem

Considering Eq. (1), with $F(x,t) = 0$, the Direct Problem is posed as: given the medium parameters D, V and R, the initial and boundary conditions, we must find the concentration solution $c(z)$, for all $z = (x,t) \in (0,L) \times (0,T]$.

DOI: 10.5540/03.2023.010.01.0101 © 2023 SBMAC

On the other hand, the Inverse Problem of Parameter Estimation studied here is posed as: given the medium parameters of dispersion and velocity in Eq. (1), D and V, the initial and boundary conditions g_0, g_l and g_r, and some measurements of concentrations from sensors, we are interested in determining the reaction parameter, R.

2.2 The Source Location Estimation Problem

Consider the problem (1), with

$$F(x,t) = \lambda_1 \left(e^{-(x-\lambda_2)^2} \right), \text{ with } 0 < \lambda_2 < L, \tag{2}$$

where $\lambda_1 > 0$ is the intensity of the source pollutant concentration and $\lambda_2 \in [0, L]$ is the source location. This kind of source represents a continuous punctual leakage scenario concentrated in position λ_2 and with decreasing intensity when we analyse a position far away of λ_2. The Direct Problem for (1) is posed as: given the vector of parameters $\lambda = (\lambda_1, \lambda_2)$, we must to find the solution $c_\lambda(x,t)$. Here we consider that the intensity is the same for all possible locations.

3 The Physics-Informed Neural Network Approach

As mentioned in the Introduction, the recently developed Physics-Informed Neural Network (PINN) defines an optimization process, in which the weights and biases minimizes a special modified loss function that takes the physical laws relevant to the problem in consideration.

In this section, we present the main ideas about the PINN, and how we use them to solve the inverse problem of parameter estimation and the other of source location estimation.

First, we point out the main differences between construct an usual Artificial Neural Network and a Physics-Informed Neural Network.

The ANN is composed by a collection of neurons (or nodes) distributed in layers that apply different transformations on their inputs. This neurons are weighted to adjust the training process. The aim in the training phase of an ANN is to find a set of optimal weights that minimizes the error between the prediction (output) and the target values.

The main difference between the ANN and the PINN are the target values. In the ANN, we provide a list of target values that are used for tunning the parameters of the net in the loss function of Mean Squared Error type. On the other hand, in the PINN, there are no target values because we are interested in minimizing a new loss function that incorporates the physics of the problem through the differential equation that govern the observed data [8].

Consider the both problems in the pollutant dispersion in a river, posed by Eq. (1). Observe that we have information about the pollutant concentration initially (with the function $g_0(x)$) and on the boundary (with the functions $g_l(z)$ and $g_r(z)$). Furthermore, in the inverse problem context, we can suppose that, after the pollutant leakage, we have information about the concentration from observational data provided by sensors distributed along the river. Correlated to this data from sensors there is a reaction parameter (unknown) R_s that we are interested in estimating, in the first problem, where we consider the homogeneous equation. In the second problem, correlated to a corresponding observational data from sensors, there is a (unknown) source location λ_s, that we are interested in estimating, where we conside the non-homogeneous equation.

Let us take into account an ANN whose inputs are the nodes (z_i), for $i = 1, 2, \ldots, N_d$. We observe that the set of points $\{z_i\}_{i=1}^{N_d}$ can be seen as the collection:

$$\{z_i\}_{i=1}^{N_d} = \left\{z_i^0\right\}_{i=1}^{N_0} \bigcup \left\{z_i^b\right\}_{i=1}^{N_b} \bigcup \left\{z_i^s\right\}_{i=1}^{N_s}, \tag{3}$$

 © 2023 SBMAC

where N_0, N_b and N_s correspond to the number of domain points that generate the initial condition, boundary conditions and sensor measurements, respectively. So, the total number of domain points, N_d, can be view as the sum $N_d = N_0 + N_b + N_s$. Observe that the splitting in Eq.(3) will generate different terms in the loss function definition. Besides, the targets are the corresponding concentrations already known by the problem, that is, the initial condition $c(z_i^0) = c(x_i^0, 0) = g_0(x_i)$, for $i = 1, 2, \dots N_0$, the boundary conditions

$$c(z_i^b) = c(x_i^b, t_i^b) = \begin{cases} g_l(t_i^b), \text{ if } x_i^b = 0 \\ g_r(t_i^b), \text{ if } x_i^b = L, \end{cases}$$

and the sensors measurements $c_s(z_i^s, R_s)$, for $i = 1, 2, \dots N_s$ (analogously to the case $c_s(z_i^s, \lambda_s)$) . The output of this ANN is the function $c_{NN}(z_i, \theta)$, where θ is the vector of all parameters of the network.

Therefore, the loss function, $L_d(\theta)$, of Mean Squared Error (MSE) type for the ANN, uses only the data from the problem and from the sensors and is given by $L_d(\theta) = L_0(\theta) + L_b(\theta) + L_s(\theta)$, that is,

$$L_d(\theta) = \frac{1}{N_0}\sum_{i=1}^{N_0} |c_{NN}(z_i^0, \theta) - g_0(x_i)|^2 + \frac{1}{N_b}\sum_{i=1}^{N_b} |c_{NN}(z_i^b, \theta) - c(z_i^b)|^2 + \frac{1}{N_s}\sum_{i=1}^{N_s} |c_{NN}(z_i^s, \theta) - c_s(z_i^s, R_s)|^2. \tag{4}$$

For the purpose of defining the new loss function for the PINN case, we need to define the following functional, namely residue of Eq. (1):

$$\mathcal{R}(u) := \partial_t u - D\partial_{xx} u + V\partial_x u + Ru - F(x, t), \tag{5}$$

where u is an appropriated function. In the first problem, the corresponding functional is given by $\mathcal{R}_1(u, R)$, with $F(x, t) = 0$ and $R > 0$ a constant. The second problem, we have the corresponding functional given by $\mathcal{R}_2(u, \lambda)$, with $F(x, t) = e^{-(x-\lambda)^2}$ and λ a constant. In a general way, we note that if $\mathcal{R}(c) = 0$, then we say that $u = c$ is a solution (general) of the PDE in Eq. (1), with known values for the parameters D, V, R and λ.

Since we are approximating the pollutant concentration solution by a neural network $c_{NN}(z, \theta)$, then it will result in a different approximation by a neural network for $\mathcal{R}$, namely a Physics-Informed Neural Network (PINN) [8]. This method involves training a neural network to approximate the concentration data in Eq.(4), and, simultaneously, the solution of the PDE in Eq. (1) by minimizing a new loss function that incorporates the residue given by Eq.(5) in a new summation term.

It is important to point out that we are assuming that both in the neural networks' output c_{NN} and in the residue $\mathcal{R}(c_{NN}, R)$ (for first problem) or $\mathcal{R}(c_{NN}, \lambda)$ (for second problem), the set of parameters θ is the same. So, in the first problem, for example, the parameters (θ, R) can be optimized by minimizing the new loss function, $L(\theta, R)$, defined by $L(\theta, R) = L_d(\theta) + L_r(\theta, R)$, where $L_d(\theta)$ is the loss term due to the data, given by Eq.(4), and $L_r(\theta, R)$ is the loss term due to the residue, defined by

$$L_r(\theta, R) = \frac{1}{N_r}\sum_{i=1}^{N_r} |\mathcal{R}_1(c_{NN}(z_i^r, \theta), R)|^2, \tag{6}$$

where $\mathcal{R}_1(c_{NN}(z_i^r, \theta), R)$ is given by (5) for the first problem, substituting u by $c_{NN}(z_i^r, \theta)$ and $\{z_i^r\}_{i=1}^{N_r}$ are collocation points distributed over the entire domain. In a similar way, we can define the new loss term $L(\theta, \lambda)$ for the second problem.

We observe, that for the PINN case, we append a new set of points: the collocation points. The collocation points are chosen so that the approximation to the solution satisfies the differential

 © 2023 SBMAC

equation at those points. With this new set of points, we have the following

$$\{z_i\}_{i=1}^{N_d} = \{z_i^0\}_{i=1}^{N_0} \bigcup \{z_i^b\}_{i=1}^{N_b} \bigcup \{z_i^s\}_{i=1}^{N_s} \bigcup \{z_i^r\}_{i=1}^{N_r}. \tag{7}$$

This approach for the construction of the loss function is slightly different from that presented in [8], in which it only considers the data from sensor measurements in the loss term due to the data. In this work, we consider, besides the sensor measurements, the data from initial and boundary conditions in the corresponding loss term.

4 Numerical Experiments

In this section, we present numerical experiments related to both problems, the reaction parameter estimation and the source location estimation with PINN. We consider the same PDE parameters, domain and initial and boundary conditions in the two problems. The PINN was implemented in the Python language using the well known open-source library for machine learning, TensorFlow [1]. However, in view of limited CPU, we choose to use the Google Colab, which is a way of writing and executing Python codes directly in the internet browser, even without any Python software installed in the computer, and with access to GPUs free of charges in a free account[5].

4.1 The Reaction Parameter Estimation

In this subsection, we implement the reaction parameter estimation with PINN. In Tables 1 and 2, we consider only the network with 4 hidden layers because this architecture generated better results.

In Table 1, we consider the parameter estimation problem solved by different number of neurons, activation function as the Leaky RELu function, with slope $\alpha = 0.1$, $N_x = 50$ and $N_t = 50$ equally spaced points in the entire domain $[0, L] \times [0, T]$, and $N_r = 1000$ collocation points sampled from the domain. In all experiments, we use ADAM optimizer [6], with learning rate varying accordingly of number of epochs in the following way $\delta(n) = 0.1(1 - H(n - 2000)) + 0.01(H(n-2000) - H(n-4000)) + 0.001H(n-4000)$, where $H(n) = 0$, for $n < 0$, and $H(n) = 1$, for $n \geq 0$, is the Heaviside or step function. The number of epochs is fixed in 5000 and the initial guess of $R^{(0)} = 0.5\, s^{-1}$, the median value of the range considered for the reaction parameter $[0, 1\, s^{-1}]$. We observe that the best estimation was obtained considering 40 neurons. Therefore, in the next experiment, we will consider only networks of 4 layers with 40 neurons.

Table 1: Estimation of the Reaction Parameter with the PINN - 4 Layers and variable number of neurons - Noiseless data - $N_s = 50$, $N_0 = N_b = 50$, $N_r = 1000$

Layers	Neurons	Exact Parameter	Predicted Parameter	Standard Deviation	Rel. Error
4	20	0.3	0.25128	0.14861	16.24 %
4	30	0.3	0.33838	0.029058	12.79 %
4	40	0.3	0.29732	0.045473	0.89 %
4	50	0.3	0.18453	0.032688	38.49 %

In Table 2, we consider noisy measurements. The noise, σ, is introduced here in the same way as in the ANN case, that is, $c_s(z^s, R_s) + \epsilon$, $\epsilon \sim N(0, \sigma^2)$.

[5] https://colab.research.google.com/

 © 2023 SBMAC

Table 2: Estimation of the Reaction Parameter with the PINN - 4 Layers and 40 neurons - Data with different levels of noise - $N_s = 50$, $N_0 = N_b = 50$, $N_r = 1000$

Noise	Exact Parameter	Predicted Parameter	Standard Deviation	Rel. Error
0.0 %	0.3	0.29732	0.045473	0.89 %
0.1 %	0.3	0.30550	0.056082	1.83 %
0.5 %	0.3	0.30988	0.031200	3.29 %
1 %	0.3	0.26628	0.049348	11.24 %

4.2 The Source Location Estimation

In this subsection, we implement the source location estimation from the PINN. In the following tables, we only consider networks with 4 layers, in view of this number of layers already generated good results. We also consider as activation functions the Leaky RELu, with slope $\alpha = 0.1$. In all experiments, we use ADAM optimizer with learning rate varying accordingly of epoch of the following way $\delta(n) = 0.01(1-H(n-1000))+0.001(H(n-1000)-H(n-2000))+0.0001H(n-2000)$. The number of epoch is fixed in 3000 and the initial guess of $\lambda_2 = 5.0$.

In Tables 3 and 1, we consider different numbers of measurements from sensors and different numbers of neurons to verify the accuracy of the method.

In the Table 3, we sampled 1000 collocation points from the entire domain and 50 points from the x-direction and 50 from t-direction. Beside this, we note that the network architecture with 4 layers and 40 neurons generated better results.

Table 3: Estimation of Pollutant Source Location with the PINN - 4 Layers and variable number of neurons - Noiseless data - $N_s = 50$, $N_0 = N_b = 50$, $N_r = 1000$

Layers	Neurons	Exact Location	Predicted Location	Standard Deviation	Rel. Error (%)
4	20	3.7	3.6694	0.00810	0.83
4	40	3.7	3.6777	0.05991	0.60
4	60	3.7	3.6645	0.02770	0.96

In Table 4, we keep all the features from table 3, bur now considering the noise effect. The network architecture considered here is the 4 layers with 40 neurons. We note that all the results for PINN can reconstruct the location even in presence of noise in the measurements from sensors.

Table 4: Estimation of Source Location with the PINN - 4 Layers and 40 neurons - Data with different levels of noise - $N_s = 50$, $N_0 = N_b = 50$, $N_r = 1000$

Noise (%)	Exact Location	Predicted Location	Standard Deviation	Rel. Error (%)
0.0	3.7	3.6777	0.05991	0.600
0.1	3.7	3.6779	0.00958	0.598
0.5	3.7	3.6749	0.01456	0.678
1	3.7	3.6651	0.03802	0.945

5 Conclusions

In this work, we study two inverse problems, the the reaction parameter estimation and the source location estimation in the advection-dispersion-reaction equation modelling the pollutant

 © 2023 SBMAC

concentration in a river. These inverse problems were solved by using a new kind of neural network, namely Physics-Informed Neural Networks.

The direct problems were solved by classical numerical methods with the goal of generating a dataset to be used in the neural network approach for the inverse problem.

The inverse problems were solved with the PINN. We observed that, despite considering the noisy measurements, it proved to reconstruct the reaction parameter as well as the pollutant source location both with a low relative error.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, National Council for Scientific and Technological Development - CNPq and FAPERJ - Carlos Chagas Filho Foundation for the Research Support of the State of Rio de Janeiro. H.S.M. also acknowledge the Rio de Janeiro State University (UERJ), for the PAPD Program and FAPERJ - Carlos Chagas Filho Foundation for the Research Support of the State of Rio de Janeiro for the Emeritus Visiting Researcher grant.

References

[1] M Abadi et al. **TensorFlow: Large-scale machine learning on heterogeneous systems**. Software available from tensorflow.org. 2015.

[2] M Andrle and A El Badia. "On an inverse source problem for the heat equation. Application to a pollution detection problem, II". In: **Inverse Problems in Science and Engineering** 23.3 (2015), pp. 389–412.

[3] A G Baydin et al. "Automatic Differentiation in Machine Learning: a Survey". In: **Journal of Machine Learning Research** 18 (2018), pp. 1–43.

[4] A El Badia, T Ha-Duong, and A Hamdi. "Identification of a point source in a linear advection–dispersion–reaction equation: application to a pollution source problem". In: **Inverse Problems** 21 (2005), pp. 1121–1136.

[5] L C Evans. **Partial Differential Equations**. 1st Edition. Americal Mathematical Society, 1998.

[6] D P Kingma and J L Ba. "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION". In: **Proceedings of the International Conference on Learning Representations - ICLR 2015**. San Diego, California, 2015.

[7] F. D. Moura Neto and A. J. Silva Neto. **An Introduction to Inverse Problems with Applications**. 1st Edition. New York: Springer, 2013.

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

 © 2023 SBMAC