Data-driven methods for partial diferential equations

Vishwas Rao

January 2024

The goal of this project is to develop data-driven methods to solve partial differential equations (PDEs). PDEs are using in The numerical simulation of many natural phenomena such as climate, weather, geophysical processes etc. Solving these complex PDEs are computationally expensive. Data-driven methods are becoming increasingly popular for surrogate modeling (emulation) or data analyses of all or a portion of the processes involved in numerical weather prediction, climate modeling etc. In this project we will explore machine learning (ML) based methods to solve elliptical PDEs and elliptical PDEs are used to model subsurface flows, in diffuse optical tomography, and several other systems. A simplest example of elliptic PDE's is the Poisson's problem. The following equations describe a 2D elliptic PDE:

$$-\nabla \cdot (e^u \nabla w) = f(x, y) \quad \text{in } \Omega \tag{1}$$

$$w = 0 \quad \text{on} \quad \partial\Omega \tag{2}$$

The above PDE represents an steady state elliptic heat transfer equation with Dirichlet Boundary conditions, u represents the conductivity and w represents the temperature profile over the domain. You can use $f(x, y) = 1$, $x \in [0, 1]$, $y \in [0, 1]$, and $u = e^{(-4\times(x-0.5)^2-4\times(y-0.5)^2)}$. u Just represents a Gaussian blob with center at $(0.5, 0.5)$.

At Argonne, PDEs in general and data-driven methods in particular are being used across several applications and MCS (our division) collaborates with Atmospheric and Climate scientists to develop methods that are applicable to real-life problems.

1 Description about the company

Argonne National Laboratory is a federally funded research and development center in Lemont, Illinois, United States. Founded in 1946, the laboratory is owned by the United States Department of Energy and administered by UChicago Argonne LLC of the University of Chicago. The facility is the largest national laboratory in the Midwest. The student will be hosted by the mathematics and computer science (MCS) division. A leader in the computing sciences, the MCS Division provides the numerical tools and technology for solving some of our nation's most critical scientific problems. In addition to our world-class research, we develop the software for some of the fastest, most powerful computer systems in the world: systems that are enabling scientists to tackle problems previously considered infeasible.

2 Project description

In this project we develop ML methods to solve PDEs. The goal is to implement methods such as Fourier Neural Operator, Physics Informed Neural Nets etc. to solve simple PDEs. Initially, the student will develop simple discretization schemes (Finite Difference methods) to solve the PDEs using traditional numerical methods. This exercise will also help us in gathering data for training ML algorithms. The student will be required to get familiar with the literature to understand the ML algorithms and then implement them. Most of the implementation will be done in Python.

3 Theoretical Background

Some familiarity with PDEs is required. Familiarity with finite difference methods is desriable. Experience with Python and PyTorch is necessary. All of the above material can also be learnt during the course of the internship. It will also be useful if the students get themselves familiar with the following papers: [1, 2].

4 Primary Goal

The goal is to read, understand, and implement the research papers in computational science. This is a valuable research experience and good starting point for any aspiring researcher.

5 Potential Tasks

Below, is the outline of weekly task:

1. Write down the discretization for the PDE (for an 1D-problem to make things easier) and implement in python.

2. Test and validate the 1D implementation of the PDE in python.

3. Read and understand the ML algorithm for solving PDEs (PINNs, FNOs etc).

4. Implement the ML algorithm for 1D-PDE
5. Write down the discretization for the PDE for an 2D-problem and implement in python.
6. Test and validate the 2D implementation of the PDE in python.
7. Implement the ML algorithm for 2D-PDE
8. Report writing/presentations
9. Report writing/presentations

6 Outcomes

By the end of the project, the student has gained hands-on experience of reading a research paper and implementing it. The would have faced many challenges and overcome them (either by themselves or with the help of the supervisor). This will give them the confidence to work on more challenging problems.

References

[1] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[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. *Journal of Computational physics*, 378:686–707, 2019.