Notes for: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

Seth Jones

June 18, 2023

0.1 Overview

This paper discusses the concept of Physics-Informed Neural Networks or PINNs, which is a very powerful deep learning framework that is designed to solve forward and inverse problems associated with nonlinear PDEs. This approach combines the flexibility and efficiency of neural networks with the physical constraints inherent in PDEs. This method is capable of accurately predicting the solutions to complex PDEs while simultaneously learning the underlying governing equations with limited data. This paper demonstrates the effectiveness of this method and how it can be applied to other applications which have nonlinear PDEs in them. The paper discusses how traditional methods for solving PDEs are often too computationally expensive and tend to struggle with real-world problems. This is where neural networks (NNs) have shown great potential in solving complex tasks but normally requires a large amount of data in traditional methods. This paper promises to show that PINNs offer a viable solution by using both NNs and physical laws that enable accurate solutions in problems where there is not a lot of data. This paper discusses that the architecture of the NNs that are used in PINNs are customized to incorporate the governing equations as additional constraints. By incorporating these equations into the loss function, PINNs ensure that the predicted solutions obey the underlying PDEs.

0.2 Loss Functions

The loss function is described to have multiple components, some are common in most NNs while others are unique to PINNs

- Data-Fit Loss: This loss is described to be the difference between the predicted solution and observed data which is common in most NNs.
- PDE Residual Loss: This loss enforces that the NN adheres to the PDEs at all points in the domain.
- Boundary Conditions Loss: Ensures that the NN follows the boundary conditions.

0.3 Continuous Time Models and Discrete Time Models

The authors discuss how they devised two distinct types of algorithms, continuous time and discrete time models.

0.3.1 Continuous time models

The authors make a distinction that their proposed algorithm is different from others in the literature by discussing other methods which use the term physics-informed machine learning are still using machine learning approaches like random forests, Gaussian process, ect. while this approach uses a "customized" activation and loss functions that are "tailored to the underlying differential operator.

0.3.2 Discrete time models

Like the continuous time model above, the proposed method is inserting the "underlying differential operator" into the Runge-Kutta time stepping scheme. By employing Runge-Kutta with a large number of stages, a high level of accuracy in time integration can be achieved.

0.4 Applications of PINNs

Applications that are described seem to be able to be broken down into two main categories, Forward problems and Inverse problems.

1. Forward Problems:
    (a) PINNs can accurately predict solutions to forward problems involving nonlinear PDEs, these solutions could be for problems in the areas of fluid dynamics, heat transfer, electromagnetic, ect.
    (b) PINNs outperform traditional numerical methods in terms of accuracy and computational efficiency.
2. Inverse Problems:
    (a) PINNs can tackle inverse problems where the goal is to identify parameters or functions based on observed data
    (b) The network can lean the underlying parameters and the PDE simultaneously even from limited data. This makes PINNs especially useful in cases when data is very sparse, noisy, or unreliable.

0.5 Other Notes

PINNs seem to provide a promising approach for solving forward and inverse problems that are associated with nonlinear PDEs. By incorporating physical laws into the NN architecture, PINNs can achieve high accuracy even when there is a lack of data.