Physics-Informed Neural Networks for Heat Transfer Problems - Active sensor placement

Vishal Bondili

MATH - 8070/7070 Inverse Problems

Abstract

This project focuses on implementing inverse problems using Physics-Informed Neural Networks (PINNs) and applies them to prototype heat transfer problems [1] that are difficult to solve using traditional computational methods. Specifically, the project explores the application of PINNs to solve the sensor placement problem, which involves identifying the optimal location for sensors to capture useful information about a system or environment. The report discusses the project's implementation and results, highlighting the effectiveness of PINNs in solving this challenging problem. [1]. The project is primarly uses DeepXDE [4] a python package for solving differential equations

Contents

1 Introduction

Physics-Informed Neural Networks (PINNs) have become popular in engineering due to their effectiveness in solving realistic problems that involve noisy data and partially missing physics. PINNs use automatic differentiation to evaluate differential operators without discretization errors and multitask learning to fit observed data while respecting the governing laws of physics. This study presents various applications of PINNs to prototype heat transfer problems that are challenging to tackle with traditional computational methods. These applications include forced and mixed convection with unknown thermal boundary conditions, the Stefan problem for two-phase flow, and realistic industrial applications related to power electronics. The results demonstrate that PINNs can solve ill-posed problems and bridge the gap between computational and experimental heat transfer.

Sensor placement can be considered an inverse problem, where the goal is to determine the optimal locations to place sensors to obtain the most accurate and informative measurements. This problem can be challenging due to the high dimensionality of the search space and the need to balance measurement quality and cost.

[4]

1.1 Motivation

Sensor placement problems are active research areas, here are some points on why the sensor placement problem is important and how machine learning can help:

Maximizing sensor efficiency: Sensor placement can impact the efficiency and effectiveness of a system. Placing sensors in the wrong location can result in incomplete or inaccurate data, leading to poor decisions. Machine learning can help identify the optimal sensor placement to maximize the efficiency of the system.

Cost reduction: Optimal sensor placement can reduce the cost of installation, maintenance, and operation of sensors. Machine learning algorithms can help identify the minimum number of sensors required to achieve the desired accuracy, which can save money and resources.

Improved decision-making: The accurate and timely data provided by sensors can improve decision-making processes in various domains, including healthcare, transportation, and manufacturing. Machine learning algorithms can identify the most informative sensor placements to enable effective decision-making.

Real-time monitoring: Real-time monitoring is critical in many applications, including environmental monitoring, surveillance, and process control. Optimal sensor placement is necessary to ensure timely and accurate data acquisition. Machine learning can help identify the best locations for sensors to enable real-time monitoring.

2 Physics-Informed Neural Networks (PINNs)

2.1 Overview

In the context of PINNs, a neural network is used to approximate the solutions of PDEs. The basic idea behind PINNs is to use a neural network to represent the solution of the PDE and to enforce the governing equations of the problem as constraints during training. This is done by adding a term to the loss function of the neural network that penalizes violations of the PDE. In addition to approximating the solution of the PDE, PINNs can also be used to perform a range of tasks such as parameter estimation, uncertainty quantification, and control. PINNs have been successfully applied to a variety of fields such as fluid dynamics, heat transfer, electromagnetics, and structural mechanics.

One of the advantages of using PINNs is that they can be trained using a small number of samples, making them particularly useful for problems where data is limited or expensive to obtain. However, they can be computationally expensive to train and may require careful tuning of hyperparameters.

Overall, PINNs offer a promising avenue for solving complex PDE problems in a more efficient and accurate manner, and have the potential to be used in a wide range of engineering and scientific applications.

2.2 Governing Equations

The problem considered is a forced convection problem. The governing equations of the problem are given by incompressible **Navier–Stokes equation**[2] and the corresponding **Temperature equation**

$$\frac{\partial \theta}{\partial t} + (\mathbf{u} \cdot \nabla)\theta = \frac{1}{\mathrm{Pe}} \nabla^2 \theta$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u} + \mathrm{Ri}\theta$$

$$\nabla \cdot \mathbf{u} = 0$$

where $\theta, u = (u, v)^T and p$ are the dimensionless temperature, velocity, and pressure fields, respectively. P_e, R_e, R_i denote the Peclet, Reynolds, and Richardson numbers, respectively.

3 Problem Design

3.1 Simulation Domain

The experiment is performed in a closed enclosure and is classical two-dimensional heat transfer problem of forced heat convection around a circular cylinder in steady-state. The simulation domain size is [-7.5 D,22.5 D] x [-10 D,10 D]consisting of 2094 quadrilateral elements, where D is the diameter of the cylinder. The cylinder center is located at (0,0). The Boundary conditions are implemented using Dirichlet and Neumann Boundary conditions

3.2 Boundary Conditions

On the fluid flow part, the cylinder surface is assumed to be no-slip, no-penetration wall.

Inflow Boundary: Uniform velocity $u = U_\infty, v = 0$ is imposed on the inflow boundary where x = 7.5 * D

Outflow Boundary: zero-pressure boundary is prescribed on the outflow boundary where x = 22.5 * D

Lateral Boundaries: periodic boundary condition is used on the lateral boundaries where y = - 10 * D

For **Heat Transfer part**:

Inflow Boundary: Constant temperature is imposed on the inflow boundary

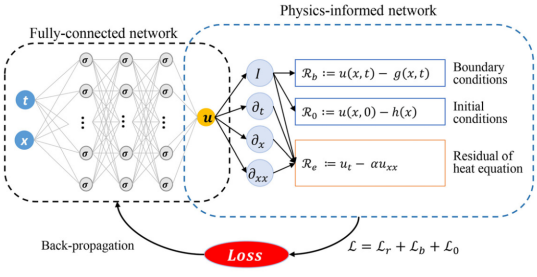
Outflow Boundary:zero-gradient is used on the outflow boundary $\delta\theta/n = 0$

Lateral Boundaries: Periodic boundary is assumed on the lateral boundaries.

4 Proposed PINN Design

4.1 Design

To solve the problem a of sensor placement a fully connected neural network is employed with ten hidden layers and 200 neurons per layer with inputs of (x, t)



To approximate the solution of $u(x, t)$ in the given problem, a fully connected neural network is utilized. This network is trained using gradient descent methods based on back-propagation of the loss function. The loss function is constructed from three different losses - the residual loss L_r, boundary conditions loss L_b, and initial conditions loss L_θ. Automatic differentiation in TensorFlow is used to compute the derivatives of u.

4.2 Active Sensor Placement

The criterion of adding sensors we propose is based on the residual of the temperature equation

Step 1: Provide an initial sensor configuration.

Step 2: Train the neural network (PINN).

Step 3: Compute the residual on the cylinder boundary inferred by PINN

Residual:

$$e = u\theta_x + v\theta_y - P_e^{-1} * (\theta_{xx} + \theta_{yy}) \quad (1)$$

Step 4: Find the position with the maximum value of residual.

Step 5: Update the sensor placement and training data, and go back to step 2.

Here the residual, which represents the error in the temperature equation can be directly computed by the neural network and it does not require any prior knowledge

We add sensor at the position with the highest residual as it represents the location with the largest error in the temperature equation. By adding a sensor at this location, the overall accuracy of the temperature prediction can be improved.

4.3 Loss Functions

The PINN loss function can be expressed as

$$L = L_r + L_{ub} + L_{\theta b} + L_\theta \tag{2}$$

where

$$L_r = \frac{1}{n_r} \sum_{k=1}^{4} \sum_{i=1}^{N_r} |e_k(x^i, y^i)|^2 \tag{3}$$

$$L_{ub} = \frac{1}{n_{ub}} \sum_{i=1}^{N_{ub}} [u(x^i, y^i) - u_b^i]^2 \tag{4}$$

$$L_{\theta b} = \frac{1}{n_{\theta b}} \sum_{i=1}^{N_{\theta b}} |\theta(x^i, y^i) - \theta_b|^2 \tag{5}$$

$$L_\theta = \frac{1}{n_\theta} \sum_{i=1}^{N_\theta} |\theta(x^i, y^i) - \theta_{data}^i|^2 \tag{6}$$

The first term L_r penalizes the governing heat equation, the momentum equations, and the continuity equation. N_r is the batch size of residual points, which are randomly selected in the spatial domain. The second and third terms are the boundary conditions for velocity and temperature fields, respectively. θ_b denotes the environmental temperature. The last term of the loss function L_θ represents the data fitting term.

To solve the partial differential equations, A fully connected neural network with ten hidden layers, each consisting of 200 neurons, is used. The loss function includes different terms that correspond to residual points, upper boundary points, and boundary points, with the number of points specified as N_r = 20,000, N_ub = 900, and $N_{\theta b}$ = 500, respectively. The number of measurements N_θ depends on case by case typically N_θ varies from 5 to 11.

To evaluate the performance of the neural network on inferring temperature, velocity, and pressure fields it is quantitatively evaluvated against the numerical reference solution which can be computed using methods like spectral element method. The Spectral Element Method (SEM)[3] is a numerical technique for solving partial differential equations (PDEs) in complex geometries. It combines the accuracy of spectral methods with the flexibility of finite element methods, making it particularly well-suited for problems involving irregular or complex geometries.

we use the relative L_2 errors as the metric.

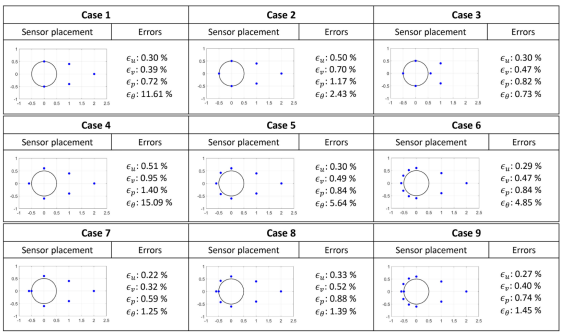
$$\epsilon_V = \frac{||V - V^*||_2}{||V^*||_2} \tag{7}$$

V represents one of the predicted quantities (θ, u, v, p), and V^* represents the corresponding reference.

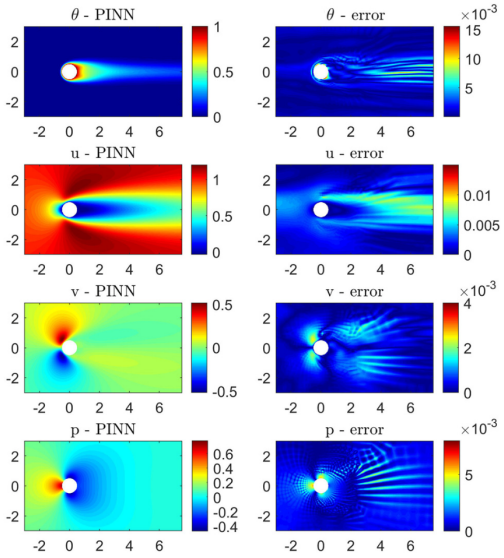
5 Results

5.1 Results

The optimization problem involved in training network parameters for PINN simulations is nonconvex, which means there is no guarantee of finding a global minimum. This can result in variability in the outcome of each simulation due to the randomness in network initialization. To account for this, ten separate training processes for each configuration are conducted and best results are reported.



Different results obtained from different sensor configurations are shown above. Nine different sensor configurations are considered for the case. The relative L_2 errors of temperature, velocity, and pressure fields are computed over the whole domain along with the sensor placements. Case 3 has the least L_2 error out of all the Nine different configurations.



The above figure shows an example of the results obtained through PINN inference, where the fields are inferred using sensor configuration case 3. The figure also includes the pointwise absolute errors between the CFD simulation and the PINN results. These errors are relatively small compared to the magnitudes of the inferred quantities.

5.2 Applications

PINNs in heat transfer problems have major applications in power electronics. Below are few of the applications

Predicting heat transfer coefficients: PINNs can be used to predict the heat transfer coefficients for different boundary conditions and geometries.

Solving inverse heat conduction problems: PINNs can be applied to solve inverse heat conduction problems, which involve determining the unknown heat flux or temperature distribution from the measured temperature or heat flux data.

Design optimization of heat exchangers: PINNs can be used to optimize the design of heat exchangers by predicting the temperature distribution and heat transfer rates under different design configurations.

Control of thermal systems: PINNs can be utilized for the control of thermal systems by predicting the temperature distribution and heat transfer rates under different control strategies.

Analysis of thermal energy storage systems: PINNs can be employed to model and optimize thermal energy storage systems, which are critical components of renewable energy systems.

5.3 Conclusion

The conventional data-driven approach used in neural networks is not suitable for heat transfer problems because of limited data availability and challenges associated with measuring temperature and velocity fields that can vary quickly over space and time. However, PINNs provide a hybrid model that allows us to use any available data at any location and time and enforce the governing equations using automatic differentiation at random points without relying on complex and costly mesh generation. This makes PINNs an attractive alternative to traditional methods in heat transfer simulations

References

[1] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-Informed Neural Networks for Heat Transfer Problems. *Journal of Heat Transfer*, 143(6), 04 2021. 060801.

[2] Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. NSFnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. *Journal of Computational Physics*, 426:109951, feb 2021.

[3] George Karniadakis and Spencer Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 06 2005.

[4] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.