PINN Application in Wave Simulation

Raghav Madan (2430477) under Prof. Subhajit Roy

Indian Institute of Technology Kanpur SURGE 2024

What are PINNs?

Traditional physics model creation is a task of a domain expert, who parametrises physics models to best fit a system of interest.
The purely data-driven neural network approach is to attempt to learn the model using supervised learning with a neural network from data obtained from a specific system.
Physics Informed Neural Networks (PINNs) lie at the intersection of the two.
PINNs uses data-driven supervised neural networks to learn the model, but also uses physics equations that are given to the model to encourage consistency with the known physics of the system.
Essentially, PINNs work on the principle that- fit the data, but make sure the solution is consistent with the physics equations that we know about

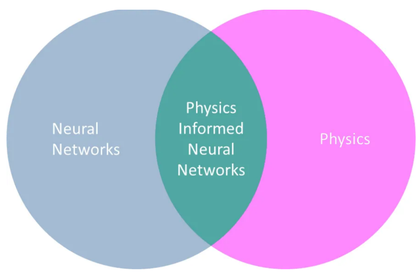


Figure 1. PINNs Illustration

They have the advantage of being both data-driven to learn a model, but also able to ensure consistency with the physics, as well as being able to extrapolate accurately beyond the available data. As such, PINNs can generate more robust models, with less data.

Key Components of PINNs

- **Neural Network Architecture**
  - A standard feedforward neural network with layers and neurons
  - In my models, I have used a neural network architecture of 2 fully connected layers of 50 neurons
- **Physical Laws**
  - PINNs need to know the physics laws/equations in order to ensure consistency with the known physics of the system
  - These laws are often represented as PDEs (partial differential equations) and are encoded as part of loss function
- **Loss function**
  - Unlike the case of data driven neural networks, the loss function of a PINN consists of data-driven loss as well as the physics informed loss
  - The physics informed loss can also be called as "PDE loss" as the physics loss term attempts to minimize the error between the gradients of the network and the corresponding equations given by known physics

More about PINNs and their advantages

Having an access to perfect data across the entire domain is rarely achievable. More realistically, we have access to data that is noisy, sparse, and incomplete.
Including the physics loss in the loss function allows the network to both regularise through data-points, as well as extrapolate outside of the training data in a manner consistent with the known physics.
PINNs can learn from the physics of the problem, allowing for more robust and accurate predictions, especially in scenarios where labeled data is scarce or costly to obtain.

Why to Involve Neural networks at all?

If we are able to learn the values for the parameters of the physics functions, why would we even need to involve neural networks at all?Why not just use gradient descent directly on the physics functions?

The main reason behind this is that as the PDEs are limited by their parametrisation, the error gradient is likely to be non-monotonic. This means that there is a high probability that the learning will converge to a local optima rather than a global one.
Many physical systems described by differential equations (like heat transfer, projectile motion) have solutions that involve undetermined parameters or constants. These parameters could represent initial conditions, boundary conditions, or other factors that are not directly captured by the differential equation alone. Therefore we do need to incorporate data loss as well

Objectives

The present study investigates the following objectives:

- Objective 1: Apply PINNs on one dimensional heat equation and predict the temperature distribution. Also observe the changes on changing the weights of physics loss term and data loss term.
- Objective 2: Predict the value of the unknown thermal diffusivity constant of the material using PINNs while predicting the temperature distribution.

Some details about my model

The partial differential equation describing the distribution of heat(or temperature) in a region (1 Dimensional) over time is given by: $\frac{\partial U}{\partial t} = \alpha \frac{\partial^2 U}{\partial x^2}$ where t is time , x is distance, U is the temperature and α is the thermal diffusivity constant.
In my model, the training data is generated using random sampling and using the closed form solution of the one dimensional heat equation.
I have used a neural network architecture of 2 fully connected layers of 50 neurons, using Gaussian Error Linear Unit (GELU) activations since GELU provides much more grounded theoretical and empirical benefits. We didn't use the standard ReLU activation because we are minimising a loss on the gradients of the network and the activation function needs to be differentiable everywhere.

Loss Functions

I have used standard loss of Mean Squared Error(MSE) between the regression targets and predicted values

$$\text{Physics Loss: } \frac{1}{n}\sum_{i=1}^{n}(\frac{\partial U_i}{\partial t_i} - \alpha \frac{\partial^2 U_i}{\partial x_i^2})^2$$

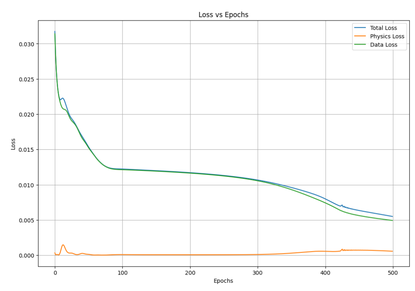
$$\text{Data loss: } \frac{1}{n}\sum_{i=1}^{n}(u_{pred_i} - u_{actual_i})^2$$

$$\text{Total Loss: } L_{physics} + L_{data}$$

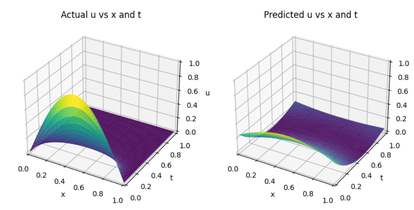
Results and discussion

Objective 1 Model results

The results obtained on applying PINN to 1D heat equation such that the weights of the physics loss and data loss are kept same are demonstrated below:



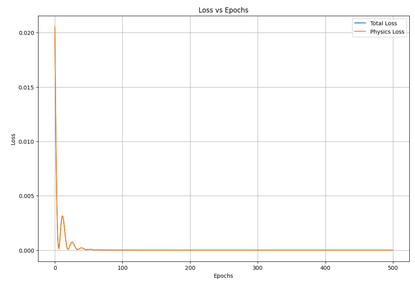
(a) Loss vs epochs



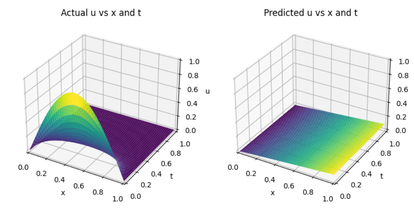
(b) Predicted vs actual temperature distribution

Figure 2. Results when weights of physics loss and data loss are equal

When we set the weight of the data loss to 0 i.e only use the physics loss to predict the temperature distribution , the results shown below are quite inaccurate



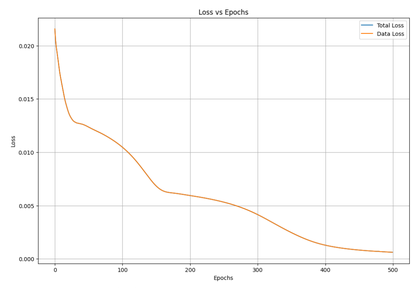
(a) Loss vs epochs



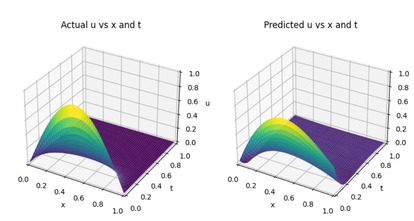
(b) Predicted vs actual temperature distribution

Figure 3. Results when we use only physics loss to predict temperature

The reason for the inaccuracies in predicting temperature distribution using only physics loss is same as the answer of "Why to involve Neural Networks at all ?"
Even though the model correctly predicts the differential equation using physics loss ,but the solution to the differential equation might be a family of equations ,with unknown constants, so model is randomly assuming the constants since we are focusing on only minimizing the physics loss.Adding some data loss will help to increase the accuracy as then model could predict the undetermined parameters using data points.
Now using purely data loss yields a very accurate model since in our case there is almost no noise in the data. With a noisy or incomplete data ,using PINNs is the best choice.



(a) Loss vs epochs



(b) Predicted vs actual temperature distribution

Figure 4. Results when we use only physics loss to predict temperature

Objective 2 Model Results

This model predicts the unknown thermal diffusivity constant using PINNs.
The model first predicts the temperature distribution as a function of distance and time. Using the assumption that the result resembles closely with the physics laws, since we used a PINN , we say that the the values generated must follow the one dimensional heat equation i.e we must have $\alpha = \frac{\partial U}{\partial t} / \frac{\partial^2 U}{\partial x^2}$
Using this we predict alpha at all points of the testing data and Plot it. The plot of predicted thermal diffusivity when the actual thermal diffusivity is 0.75 is shown below -

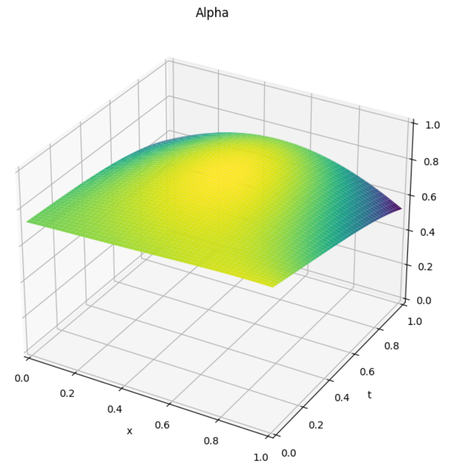


Figure 5. Predicted thermal diffusivity constant when actual is 0.75

Conclusions

- PINNs represent a powerful approach for solving complex physical problems by integrating traditional machine learning techniques with fundamental physical principles
- The best results are produced when both physics loss and data loss are considered.Using only physics loss may lead to wrong results as there could be undetermined parameters or constants

References

- Towards data science blog-"Physics Informed Neural Networks (PINNs): An Intuitive Guide"
- Kaggle guide on Physics Informed Neural Networks