Introduction to the project

Hi everyone!

Many of us have encountered differential equations during JEE. It might be the underlying differential equation (DE) for conservation of mass, motion of a spring, etc. Some of them have exact/analytical solutions.

NEED:

On the other hand, some differential equations do not have analytical solutions, requiring us to use alternative methods.

One common approach is numerical methods, which involve iterative formulas to approximate solutions.

However, the accuracy of the numerical solution can be affected by various factors (non-linearity, dimensionality, etc.) and convergence to the correct solution is not always guaranteed. Here is where PINNs come in.

NN Usage In solving DE :

Neural networks are networks of interconnected nodes. Each edge in the network represents a mathematical operation applied to the input data. Some nodes, known as activation functions, introduce non-linearity to the model (you'll learn more about these in week 2's resources).

The goal of a neural network is to approximate the desired output as closely as possible. It achieves this by calculating the error, or loss, and adjusting the weights and biases, which are the network's parameters, to minimize this error.

PINN's Solving DE :

The problem with NN's is that they overfit the data **sometimes** without understanding the **underlying physical laws** that govern our world, for eg: **conservation of mass, energy**, etc. Thus, PINNs are an effort towards giving the NN information about the physical laws and the boundary conditions (if any). This is done so by introducing another loss term. Neural networks usually optimise the loss quickly by a technique named gradient descent which uses autograd (you can learn more about this here if you want to). The same autograd is used in PINNs on a sample of training points to calculate the derivatives for the DE loss term. **Do read week 1's content to understand this in more detail.**

[In some cases, boundary conditions are also added to the model with another loss term.]

The first article is a great introduction to understanding PINNs and you can find the code on GITHUB repo of solving DE using NN and PINN .

However, its totally okay if you don't understand the code at this point. If you are not familiar with python, do get started on it, learn to set up your environment, using jupyter notebooks, etc - there are plenty of resources online.

I hope this helps you understand what we'll be working on and please do let us know if there are any doubts, we can organise a google meet and discuss. If anybody else wants to share resources, please feel free to add to this post!