# Physics Informed Neural Networks for solving Partial Differential Equations


*Diganta Samanta(21PH40020)*
*Under Guidance of*
*Dr. Vishwanath Shukla*
***Indian Institute of Technology Kharagpur***


MSc Project
Monday 28th November, 2022

# Contents

# Introduction

- **Machine Learning(ML)** is a sub-field of Artificial Intelligence.
- ML has a goal to develop algorithms that can learn from data automatically.
- **Artificial Neural Network(ANN), Neural Network(NN), or Neural Net** is a model of Machine Learning which is inspired by **neurons in the human brain**.
- **Physics Informed Neural Network(PINN)** is an application of ANN for solving Problems in physics.

# Artificial Neural Network

- The primary component of an ANN is **’stylized neurons’**.
- A neuron consists of a **linear transformation** followed by a **non-linear activation function**. There are different types of Non -linear activation functions like Perceptrons, Sigmoid, Tanh, ReLU, ELU, etc.
- ANN consists of such neurons stacked in layers. A deep Neural Network(DNN) has more than two hidden layers.

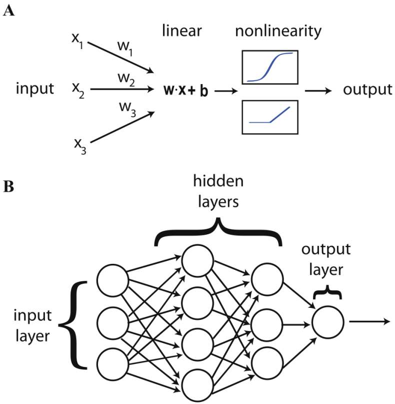


**Figure:** structure of Neural Network (Source: P. Mehta, M. Bukov, C.-H. Wang et al. / Physics Reports 810 (2019) 1–124)

# Artificial Neural Network

- **Universal Approximation Theorem:** A neural network with single hidden layer can approximate any continuous, multi-input/multi-output function with arbitrary accuracy.

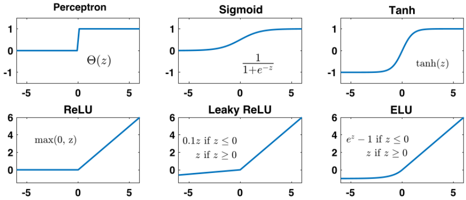


**Figure:** Some Non-linear Activation Functions (Source: P. Mehta, M. Bukov, C.-H. Wang et al. / Physics Reports 810 (2019) 1–124)

# Physics Informed Neural Network

- **Physics Informed Neural Network(PINN)** is a technique which uses Neural Network as a solution of a Partial Differential Equation.
- PINN approximates PDE solutions by minimizing a loss function that reflects the PDE, Boundary condition, Initial Condition, constraints etc.
- PINN has four parts
  1. Neural Network Architecture
  2. Auto Differentiation
  3. Loss Function
  4. Optimizer

# PINN Building Blocks

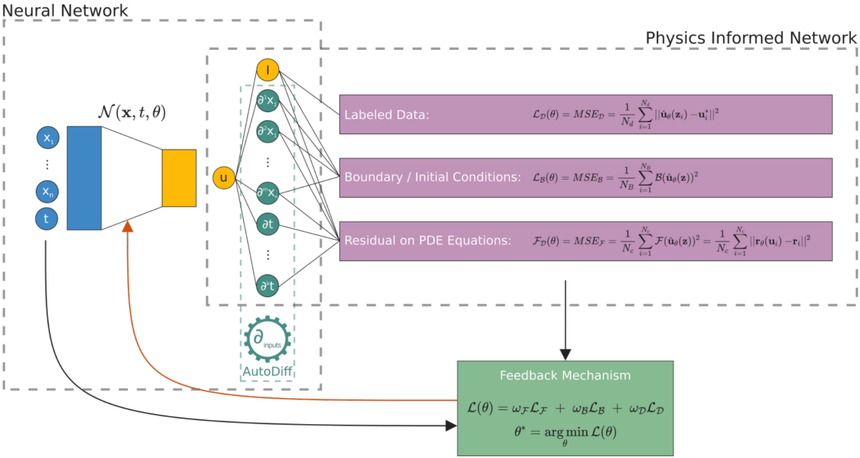


**Figure:** Building Block of PINN (source:Scientific Machine Learning through Physics-Informed Neural Networks,S Cuomo, V S di Cola, F Giampaolo, G Rozza, M Raissi, F Piccialli arXiv:2201.05624 )

# Solving Burgers' Equation

- I have solved **Burgers' Equation**

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \qquad x \in [-1, 1], \quad t \in [0, 1]$$

- Dirichlet boundary conditions and initial conditions are

$$u(-1, t) = u(1, t) = 0, \quad u(x, 0) = -\sin \pi x.$$

- I have solved for four values of viscosity $(\nu = 0.1, 0.01, 0.001, 0.0001)$

# Results for Burgers' Equation

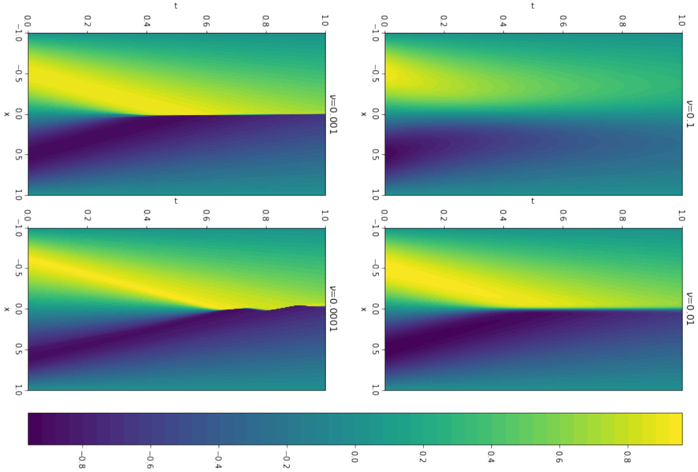


**Figure:** Solution plots of Burgers' Equation

# Solving Heat Equation

- I have solved **1d Heat Equation**

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad x \in [0,1], \quad t \in [0,1]$$

- Dirichlet boundary conditions:

$$u(0,t) = u(1,t) = 0$$

- I have solved for two initial conditions
  - Sinusoidal initial condition( taking $\alpha = 0.4$)

$$u(x,0) = \sin \pi x$$

  - Dirac Delta initial condition( taking $\alpha = 0.008$)

$$u(x,0) = \delta(x - \frac{L}{2})$$

  Where length of the rod = L = 1. I have generated Dirac function by using Gaussian distribution with variance very less than one.

# Results for Heat Equation

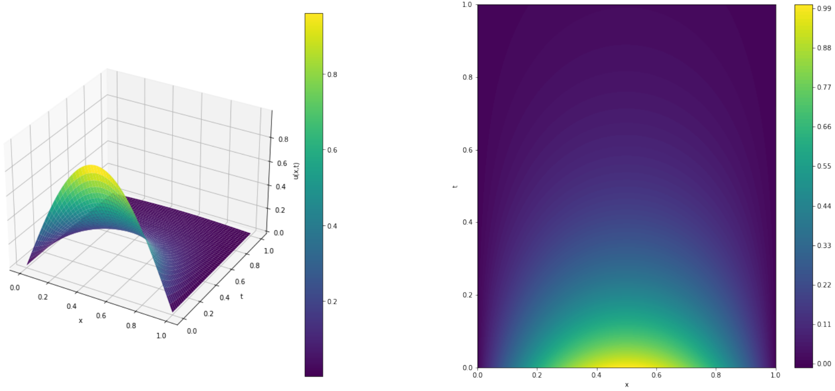


**Figure:** Plots of solution of Heat equation for periodic initial function

# Results for Heat Equation

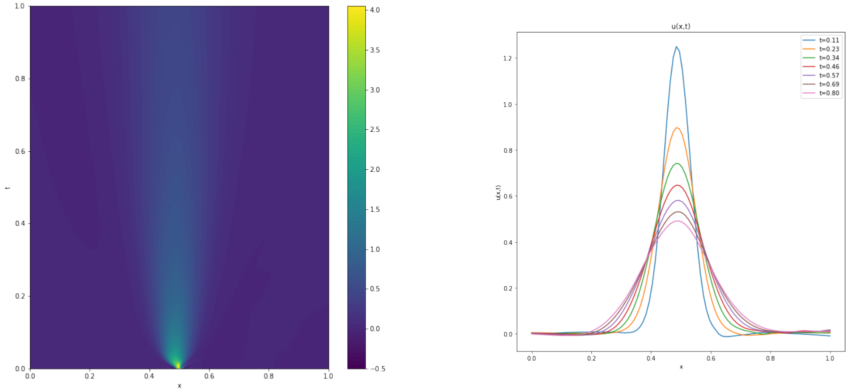


**Figure:** Plots of solution of Heat equation for Dirac delta initial function

# Advantages of PINN

- A trained PINN can predict the values on the simulation grid of **different resolutions** without being retrained.
- A trained PINN can predict the values on the simulation grid of different resolutions without being retrained. PINN uses **automatic differentiation** to compute required derivatives which is superior to numerical or symbolical differentiation.
- Besides using experimental data, PINN uses the underlying physics during the network training. So
  - If the **size of the data is small**, it can give an accurate solution.
  - It can predict a precise value if the **boundary conditions are not provided**, and there is sufficient data.

# References


1. Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next.-Salvatore Cuomo1 , Vincenzo Schiano Di Cola, Fabio Giampaolo , Gianluigi Rozza , Maziar Raissi and Francesco Piccialli
2. Lu, Lu and Meng, Xuhui and Mao, Zhiping and Karniadakis, George Em, DeepXDE: A deep learning library for solving differential equations
3. P. Mehta, M. Bukov, C.-H. Wang et al. / Physics Reports 810 (2019) 1–124)
4. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations-Maziar Raissi, Paris Perdikaris and George Em Karniadakis

# DeepXDE

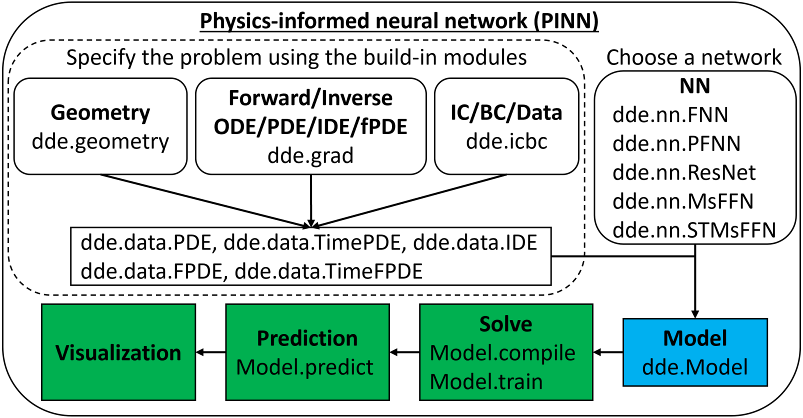


**Figure:** Implementation of PINN by DeepXDE (source:Lu, Lu and Meng, Xuhui and Mao, Zhiping and Karniadakis, George Em, DeepXDE: A deep learning library for solving differential equations)

# Training data for Burgers' equation

Network=[2,50,50,50,1], activation = 'tanh'

| $\nu$ | Optimizer | Training time (second) | final train loss |
|---|---|---|---|
| 0.1 | L-BFGS | 128.997 | $2.05 \times 10^{-06}$ |
| 0.01 | L-BFGS | 214.848 | $4.50 \times 10^{-06}$ |
| 0.001 | adam(lr=0.001) + L-BFGS | 1417.553 | $2.05 \times 10^{-06}$ |
| 0.0001 | adam(lr=0.001) + L-BFGS | 1166.165 | $2.89 \times 10^{-02}$ |

**Table:** Training times and final losses for different viscosity

# Training data for heat equation

Network = [2,20,20,20,1], activation = 'tanh'

| Initial function | Optimizer | Training time (second) | final train loss |
|---|---|---|---|
| Periodic | adam(lr=0.001) + L-BFGS | 145.807 | $6.72 \times 10^{-07}$ |
| Dirac delta | adam(lr=0.001) + L-BFGS | 145.332 | $3.60 \times 10^{-02}$ |

**Table:** Training times and final losses for different initial function

# Acknowledgements

I thank Professor Vishwanath Shukla for giving me an opportunity to work, despite my lacking of the detailed theoretical nuances needed. This project helped me to practice and sharpen my skills in the field of physics and computing.

I especially thank Dr. Abhay Kumar Tiwari, a Data Scientist at More Retail Private Ltd., who has given me valuable advice in this Deep Learning field.

I was also helped by my fellow members of StatFluid Lab who have listened to my problems and helped me when I was struck. I am grateful for their generous aid.

I want to thank also my parents, friends, roommates who have continuously motivated me.

Thank You

भारतीय प्रौद्योगिकी संस्थान खड़गपुर

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR