Machine Learning and Physics-Informed Neural Network(PINN)

Presented By: Shahab Nadir
Supervisor: Dr. Babar Zaman

Faculty of Basic Sciences
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology Topi, Swabi(KPK)

October 19, 2025

Contents

Introduction

- Imagine AI solving complex problems in science and engineering.
- Discover underlying physics from data, without numerical integration or discretization.
- Deep learning models behavior from subatomic particles to celestial bodies.
- Physics-Informed Neural Networks (PINNs) can solve non-linear PDEs.
- Non-linear PDEs are essential in Physics, Engineering, Biology, and Finance.
- Despite advanced techniques, solving non-linear PDEs is still challenging—PINNs offer new possibilities.

What is Machine Learning?

Definition

Machine Learning allows computers to learn and improve their performance on tasks through experience.

Supervised Learning

- Trained on a labeled dataset where each example is paired with an output label.
- Applications:
  - Classification (e.g., spam detection)
  - Regression (e.g., predicting house prices)
- Examples: Decision Trees, SVM, Neural Networks, Linear Regression.

Unsupervised Learning

- Trained on data without explicit labels.
- Goal: Discover hidden patterns or structures.
- Applications:
  - Clustering (e.g., customer segmentation)
  - Dimensionality reduction (e.g., PCA)
  - Anomaly detection

Reinforcement Learning

- The model (agent) learns by interacting with the environment.
- Receives rewards or penalties to maximize long-term reward.
- Applications:
  - Game playing (e.g., AlphaGo)
  - Robotics (e.g., walking or grasping)
  - Self-driving cars
- Examples: Q-learning, DQN, PPO

Deep Neural Networks (DNN)

- Multi-layered models that learn complex relationships from data.
- Key Features:
  - Layers of interconnected neurons.
  - Activation functions and backpropagation for training.
- Role in Supervised Learning: Extract features and handle high-dimensional data.

Other ML Algorithms

- Linear Models
- Tree-Based Models
- Support Vector Machines (SVM)
- Ensemble Learning
- Clustering Algorithms
- Bayesian Models
- Dimensionality Reduction Techniques

What is a PINN?

- Combines Neural Networks with physical laws (differential equations).
- Encodes governing equations (PDEs) into the loss function.
- Enables data-driven learning while respecting physics constraints.

PINN Architecture

- Fully connected feedforward neural network:
  - Inputs: (x, t)
  - Output: $u(x, t; \theta)$ (e.g., temperature)

Example: 1D Heat Equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Mathematical Formulation

$$F\left(u(x,t), \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial x^2}, \ldots\right) = 0$$

- PINN approximates the solution $u(x,t)$ satisfying the PDE above.

Loss Functions in PINNs

1. **Physics Loss (PDE Residual):**

$$L_{\text{physics}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| F\left(u(x_i, t_i), \frac{\partial u}{\partial t}(x_i, t_i) - \alpha \frac{\partial^2 u}{\partial x^2}(x_i, t_i) \right) \right|^2$$

2. **Boundary Loss:**

$$L_{\text{boundary}} = \frac{1}{N_b} \sum_{j=1}^{N_b} |u(x_b, t_j; \theta) - g(x_b, t_j)|^2$$

3. **Initial Condition Loss:**

$$L_{\text{initial}} = \frac{1}{N_i} \sum_{k=1}^{N_i} |u(x_k, 0; \theta) - h(x_k)|^2$$

Total Loss and Optimization

$$L_{\text{total}} = L_{\text{physics}} + L_{\text{boundary}} + L_{\text{initial}}$$

- Use optimization algorithms (Adam, SGD, etc.) to minimize L_{total} and update network parameters θ.

Conclusion

- Machine learning and deep learning are powerful modeling tools.
- PINNs integrate physical laws with data-driven approaches.
- They offer a powerful framework to solve PDEs in science and engineering.

Acknowledgement

Thank You!