Solving Partial Differential Equations using Physics-Informed Neural Networks (PINNs): An Application to Schrödinger's Equation

Arul Tripathi

Delhi, India

arultripathi@gmail.com

Abstract

Partial Differential Equations (PDEs) are fundamental to modeling physical systems in science and engineering. Traditional numerical methods for solving PDEs, such as finite difference and finite element methods, often face challenges in high-dimensional problems and require significant computational resources. Physics-Informed Neural Networks (PINNs) provide a novel approach by embedding the governing physical laws directly into the neural network's loss function. This paper explores the principles of PINNs and demonstrates their application to solving PDEs, with a focus on Schrödinger's equation, a cornerstone of quantum mechanics. We present a basic implementation of PINNs for solving the time-independent Schrödinger equation and discuss the potential of this method for tackling complex PDEs.

1. Introduction

Partial Differential Equations (PDEs) are widely used to describe physical phenomena, such as heat transfer, fluid dynamics, and quantum mechanics. Solving PDEs analytically is often infeasible, and numerical methods are typically employed. However, traditional numerical approaches can be computationally expensive, particularly for high-dimensional problems.

Physics-Informed Neural Networks (PINNs) have emerged as a powerful alternative, combining the flexibility of neural networks with the rigor of physical laws. By incorporating the governing equations into the loss function, PINNs ensure that the solution adheres to the underlying physics, even in the absence of large datasets. This paper focuses on the application of PINNs to solve PDEs, with Schrödinger's equation as a case study.

---

2. Physics-Informed Neural Networks (PINNs)

2.1 Neural Networks

Neural networks are machine learning models inspired by the human brain. They consist of layers of interconnected neurons that learn to approximate complex functions by adjusting weights and biases during training. Given an input ***x***, a neural network ***f(x;θ)*** with parameters ***θ*** approximates the output ***y***.

2.2 Incorporating Physics into Neural Networks

The key innovation of PINNs is the integration of physical laws into the neural network's loss function. Consider a general PDE of the form:

$$\boldsymbol{\mathcal{L}}(\boldsymbol{u}) = \mathbf{0} \text{ in } \Omega$$

Where $\boldsymbol{\mathcal{L}}$ is a differential operator, $\boldsymbol{u}$ is the solution, and $\boldsymbol{\Omega}$ is the domain. The boundary conditions are given by:

$$\boldsymbol{\mathcal{B}}(\boldsymbol{u}) = \mathbf{0} \ on\ \boldsymbol{\partial\Omega}$$

In PINNs, the neural network ***u(x;θ)*** is trained to minimize a loss function that includes both the PDE residual and the boundary conditions:

$$\boldsymbol{\mathcal{L}}_{\textbf{total}} = \boldsymbol{\mathcal{L}}_{\textbf{PDE}} + \boldsymbol{\mathcal{L}}_{\textbf{BC}}$$

where:

$$\boldsymbol{\mathcal{L}}_{\textbf{PDE}} = \frac{\mathbf{1}}{\boldsymbol{N_f}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N_f}} \left|\boldsymbol{\mathcal{L}}\big(\boldsymbol{u}(\boldsymbol{x_i}; \boldsymbol{\theta})\big)\right|^{2}$$

$$\boldsymbol{\mathcal{L}}_{\mathrm{B}} = \frac{\mathbf{1}}{\boldsymbol{N_b}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N_b}} \left|\boldsymbol{\mathcal{B}}\big(\boldsymbol{u}(\boldsymbol{x_i}; \boldsymbol{\theta})\big)\right|^{2}$$

Here, $\boldsymbol{N_f}$ and $\boldsymbol{N_b}$ are the number of collocation points in the domain and on the boundary, respectively.

---

3. Application to Schrödinger's Equation

3.1 Schrödinger's Equation

The time-independent Schrödinger equation is a fundamental PDE in quantum mechanics, describing the behaviour of a quantum system. In one dimension, it is given by:

$$-\frac{\hbar^2}{2m}\frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x)$$

where:

- ***ψ(x)*** is the wave function,
- ***V(x)*** is the potential energy,
- ***E*** is the energy eigenvalue,
- ***ℏ*** is the reduced Planck's constant,
- ***m*** is the mass of the particle.

For simplicity, we consider a particle in a one-dimensional infinite potential well, where ***V(x)*** = 0 for ***x ∈[0,L]*** and ***V(x)* = ∞** otherwise. The boundary conditions are:

$$\psi(0) = \psi(L) = 0$$

3.2 PINN Architecture

We construct a feedforward neural network ***ψ(x;θ)*** with one input neuron (for ***x***) and one output neuron (for ***ψ***). The network consists of several hidden layers with a chosen activation function (e.g., **tanh**).

3.3 Training the PINN

The loss function for this problem is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{BC}}$$

where:

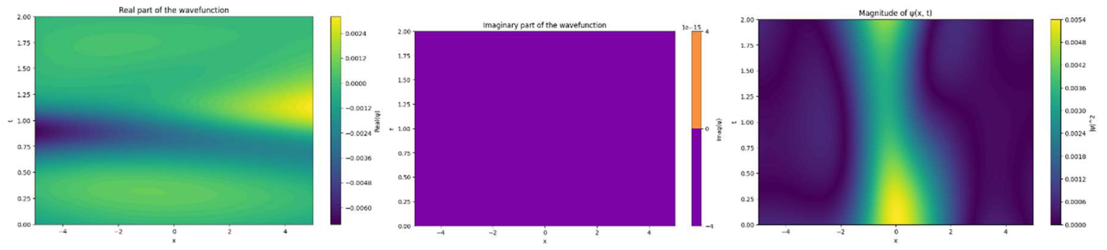
$$\mathcal{L}_{PDE} = \frac{1}{N_f}\sum_{i=1}^{N_f}\left|-\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2}(x_i;\theta) - E\psi(x_i;\theta)\right|^2$$

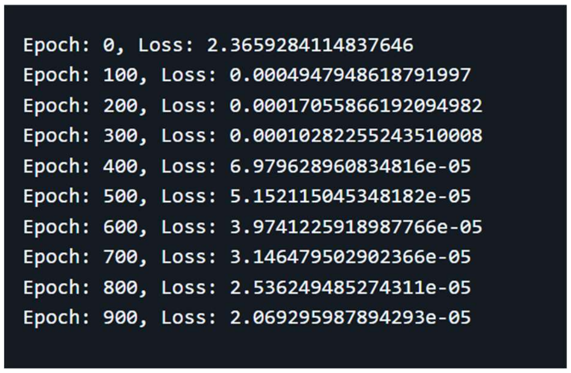
$$\mathcal{L}_{\text{B}} = \frac{1}{N_b}\sum_{i=1}^{N_b}(|\psi(0;\theta)|^2 + |\psi(L;\theta)|^2)$$

The network is trained using gradient-based optimization methods (e.g., Adam) to minimize the total loss.

3.4 Results

After training, the neural network approximates the wave function ***ψ*(*x*)** and the energy eigenvalue ***E***. The solution can be visualized and compared with the analytical solution to assess the accuracy of the PINN.





```
Epoch: 0, Loss: 2.3659284114837646
Epoch: 100, Loss: 0.0004947948618791997
Epoch: 200, Loss: 0.00017055866192094982
Epoch: 300, Loss: 0.00010282255243510008
Epoch: 400, Loss: 6.979628960834816e-05
Epoch: 500, Loss: 5.152115045348182e-05
Epoch: 600, Loss: 3.9741225918987766e-05
Epoch: 700, Loss: 3.146479502902366e-05
Epoch: 800, Loss: 2.536249485274311e-05
Epoch: 900, Loss: 2.069295987894293e-05
```

And at last, here are the results when we finally run our model.

4. Discussion

PINNs provide a flexible and powerful framework for solving PDEs, including Schrödinger's equation. By incorporating physical laws into the loss function, PINNs ensure that the solution adheres to the underlying physics, even in the absence of large amounts of labelled data. This approach is particularly advantageous for high-dimensional problems where traditional methods struggle.

However, PINNs also have limitations. Training can be computationally expensive, and the choice of hyperparameters (e.g., network architecture, activation functions) can significantly impact performance. Future research could focus on improving the efficiency and robustness of PINNs, as well as exploring their application to more complex quantum systems.

5. Conclusion

Physics-Informed Neural Networks (PINNs) represent a promising approach to solving Partial Differential Equations (PDEs) by integrating physical laws into the neural network's loss function. This paper demonstrated the basic principles of PINNs and their application to the time-independent Schrödinger equation. The results highlight the potential of PINNs as a powerful tool for solving complex PDEs, with applications in quantum mechanics and beyond.

References

1. Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686-707.

2. Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5), 987-1000.

3. Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1), 208-228.

4. Griffiths, D. J. (2005). *Introduction to Quantum Mechanics* (2nd ed.). Pearson Education.