Physics-Informed Neural Network for Option Pricing

Overview

A Physics-Informed Neural Network (PINN) is a type of neural network that incorporates physical laws into the training process. In this project, the PINN is trained to approximate the solution of the Black-Scholes-Merton (BSM) equation, a fundamental partial differential equation (PDE) in financial mathematics used for pricing European call options.

Features

- **Neural Network Architecture**: The network is built using TensorFlow's Keras API with dense and dropout layers for regularization.
- **Custom Loss Function**: The loss function includes the residual of the Black-Scholes-Merton equation, ensuring the solution adheres to the underlying financial model.
- **Training Process**: The network is trained using a custom training loop with the Adam optimizer.
- **Visualization**: Includes functionality to plot the predicted option prices against true values.

Installation

Ensure you have the following libraries installed:

```
TensorFlow
NumPy
Matplotlib
SciPy
```

You can install these dependencies using pip:

```
pip install tensorflow numpy matplotlib scipy
```

Code Description

Initialization

The `PINN` class is initialized with the following parameters:

- `layers`: List of neural network layers.
- `optimizer`: Optimizer for training the network.
- `r`: Risk-free interest rate.
- `sigma`: Volatility of the underlying asset.

BSM Residual

The function `bsm_residual` computes the residual of the Black-Scholes-Merton equation:

$$V_t + 0.5\sigma^2 S^2 V_{SS} + rSV_S - rV$$

Loss Function

The `loss_fn` function calculates the loss, which includes the mean squared error between the predicted and true option prices, as well as the residual of the Black-Scholes-Merton equation.

Training

The `train_step` and `train` functions handle the training process. The model is trained over a specified number of epochs, printing the loss every 100 epochs.

Prediction and Plotting

The `predict` function generates option price predictions for given stock prices and times to maturity. The `plot` function visualizes the predicted vs. true option prices.

License

This project is licensed under the MIT License - see the LICENSE file for details.

Acknowledgments

- Inspired by various works on Physics-Informed Neural Networks and financial mathematics.

- TensorFlow for providing an excellent framework for building and training neural networks.