Quantitative Finance - Option Pricing models

Winter in Data Science

Uid-29

By: **SHORYA SETHIA**

22B2725

ENGINEERING PHYSICS

Libraries Used

- **Pandas**: Used for data manipulation and analysis, providing a flexible and efficient DataFrame structure.
- **NumPy**: Used for numerical operations I did, facilitating array manipulation and mathematical functions.
- **Matplotlib**: Used matplotlib.pyplot for data visualization, enabling the creation of informative plots and charts.
- **Scikit-learn**: Used for train-test splitting and standard scaling, streamlining the preprocessing of data.
- **Keras** (part of **TensorFlow**): Used in the construction and training of Neural Networks, allowing for easy implementation of models, Sequential and Dense layers.
- **Seaborn**: The best library I discovered was Seaborn. Its's basically statistical data visualization library based on Matplotlib. Seaborn simplifies the creation of informative and attractive statistical graphics, and it includes various plots.
- **Torch (PyTorch)**: It's an machine learning library, providing a dynamic computational graph, particularly well-suited for deep learning and neural networks.

Data Exploration and Preprocessing

- The dataset, sourced from "ASIANPAINT_Dataset.xlsx," undergoes thorough exploration using Pandas. Missing values are addressed, and a new feature, 'sigma_20,' is introduced to capture varying **volatility**.
- The dataset is then **split into training and testing sets**, with features scaled using StandardScaler.

Neural Network Architecture

- The main model, call_model, is structured using **Keras Sequential**.
- It consists of densely connected layers with **ReLU activation**, concluding with a linear activation output layer.
- The model is compiled using the mean squared error loss function and the **Adam** optimizer.

Model Training and Evaluation

- The model undergoes training with specified **hyperparameters**, batch size, epochs, and learning rate.
- Training history is visualized, and the model is saved. Evaluation on the test set reveals the model's performance.

Results and Analysis

- The **actual vs. predicted values** are depicted using Matplotlib, offering a visual insight into the model's predictive capabilities.
- A linear fit line further emphasizes the alignment between actual and predicted values.

Conclusion

- Mastered the development and implementation of advanced financial models for options pricing, skilfully applying **deep learning** and **neural networks** to overcome **Black-Scholes equation** limitations
- Hands-on experience in engineering and refining Multilayer Perceptron **(MLP) models** (**MLP1, MLP2**) and Long Short-Term Memory **(LSTM) models**, honing accuracy and flexibility in options pricing
- Utilized **20**-day historical volatility as input for MLP models and incorporated the latest **20**-days of underlying price for the LSTM model, enabling it to dynamically learn and adapt to changing market conditions

Summary of Research Paper

Models and Architectures:

- **MLP1**: Employs a neural network with four hidden layers, improving over the Black-Scholes model in various metrics. Significantly enhances pricing accuracy for illiquid options.

- **MLP2**: Similar architecture to MLP1 but focuses on multi-task learning for bid and ask prices. Outperforms MLP1 slightly in error metrics.
- **LSTM**: Utilizes recurrent neural networks to approximate volatility from successive states. While underperforming compared to MLP models, it suggests the potential for further investigation.

Error Analysis:

- All models show superior performance to the Black-Scholes model in terms of various metrics (MSE, Bias, AAPE, MAPE, PEX%) for both call and put options.
- MLP2, emphasizing bid/ask prices, exhibits slightly improved accuracy over MLP1 in most metrics.

The research demonstrates that, even with a basic estimation of volatility, deep learning models outperform the Black-Scholes model, providing a more accurate pricing mechanism. The findings suggest a potential shift in focus towards predicting bid/ask prices for improved accuracy.

Submissions

Week 1 Assignment Submission

Week 2 Assignment Submission

No Assignment in Week 3

Week 4 Assignment (Project) Submission

GitHub Repo