Feedforward Network for Stock Market Trend Prediction

Team Members:

Mohini Patil (UMID: 32423513)

Nikhil Reddy Kandadi (UMID: 28962694)

Introduction

Stock price prediction remains a core challenge in the field of financial analysis due to the inherently volatile and non-linear nature of financial markets. This project tackles the challenge of predicting stock prices using a feedforward neural network implemented in PyTorch. The dataset includes daily closing prices of the S&P 500 and stocks from multiple sectors (airlines, finance, pharmaceuticals, and technology). The objective is to learn from historical price data to forecast the next day's price and analyze trends, correlations, and patterns.

Methodology

- **Data Preprocessing**

1. Data Preprocessing involved cleaning missing values, converting date formats, and performing exploratory data analysis (EDA) through summary statistics and line plots.
2. Correlation Analysis was conducted using Pearson coefficients to evaluate relationships between the S&P 500 and individual stocks.

- **Model Architecture**

1. The model is a feedforward neural network built using PyTorch. It consists of:
2. An input layer that accepts a vector of stock features.
3. Two hidden layers with 16 and 8 neurons respectively, each followed by ReLU activations and Dropout layers (with 10% dropout).
4. An output layer with a single neuron to predict the next day's closing price.
5. Feedforward networks are relatively simple to implement and train, serving as a good starting point before moving to more complex models.

6. Feature Extraction: - The initial layers of the feedforward network can act as feature extractors, learning relevant representations from the input data that are then used to make predictions.

❖ **Training Configuration**

1. Loss Function: Mean Squared Error (MSE)
2. Optimizer: Adam
3. Epochs: 50
4. Batch Size: 64
5. The model is trained on sliding windows of stock data, with input vectors reshaped into one-dimensional feature sets per sample.

Experiments

❖ **Trend Analysis:**

1. Line plots of stock prices over time revealed general upward trends in technology stocks (e.g., AAPL, MSFT) compared to relatively stable or volatile patterns in airline stocks (e.g., AAL, ALGT).
2. The S&P 500 index showed consistent growth throughout the year, reflecting overall market recovery.

❖ **Correlation Analysis:**

1. High correlations were observed between the S&P 500 index and major financial stocks like Goldman Sachs (GS) and Morgan Stanley (MS), indicating that these stocks closely followed market trends.
2. Technology stocks like Apple (AAPL) also showed strong positive correlations with the S&P 500.

❖ **Visualizations:**

1. The project includes several visualizations to help understand the stock trend and model predictions:
2. Line plots of actual vs predicted prices
3. Candlestick charts created with mplfinance for better visualization of OHLC data

❖ **Training Performance:**

1. The loss curve steadily declined. The model followed general price trends but struggled with rapid changes.

❖ **Limitations Observed**

1. The model tends to smooth out sharp movements due to its limited capacity to understand time-based dependencies.
2. Predictions are slightly lagging behind real values during fast uptrends or downtrends.

Conclusion

This project demonstrates that a simple feedforward neural network can capture basic patterns in stock price data and provide useful short-term forecasts. This architecture is simpler than recurrent models and is suitable for capturing patterns in flattened time windows, though it may not fully capture temporal dependencies compared to RNNs or LSTMs. While the results are promising for general trend estimation, the model may fall short when handling volatile or abrupt changes in price.

Future work could include extending the dataset to cover more years or incorporating macroeconomic indicators (e.g., interest rates, GDP growth) to deepen the analysis of sectoral performance and correlations.

Potential improvements/ Future Work include:

- Use of recurrent models like LSTM/GRU
- Hyperparameter tuning and regularization techniques for improved performance and generalization.