Time Series Prediction Using GRU Neural Network

1. Dataset Description and Preprocessing

The dataset used in this project is the Kaggle Stock Market Dataset, which contains historical daily trading data for stocks and ETFs. Each stock is stored as a separate CSV file, including attributes such as Date, Open, High, Low, Close, Adjusted Close, and Volume. For this project, a single stock was selected and only the Close price was used for prediction.

Preprocessing steps included sorting the data chronologically, applying Min-Max scaling to normalize prices to the range [0,1], and transforming the time series into fixed-length input sequences using a sliding window approach. The dataset was split into training (80%) and validation (20%) sets.

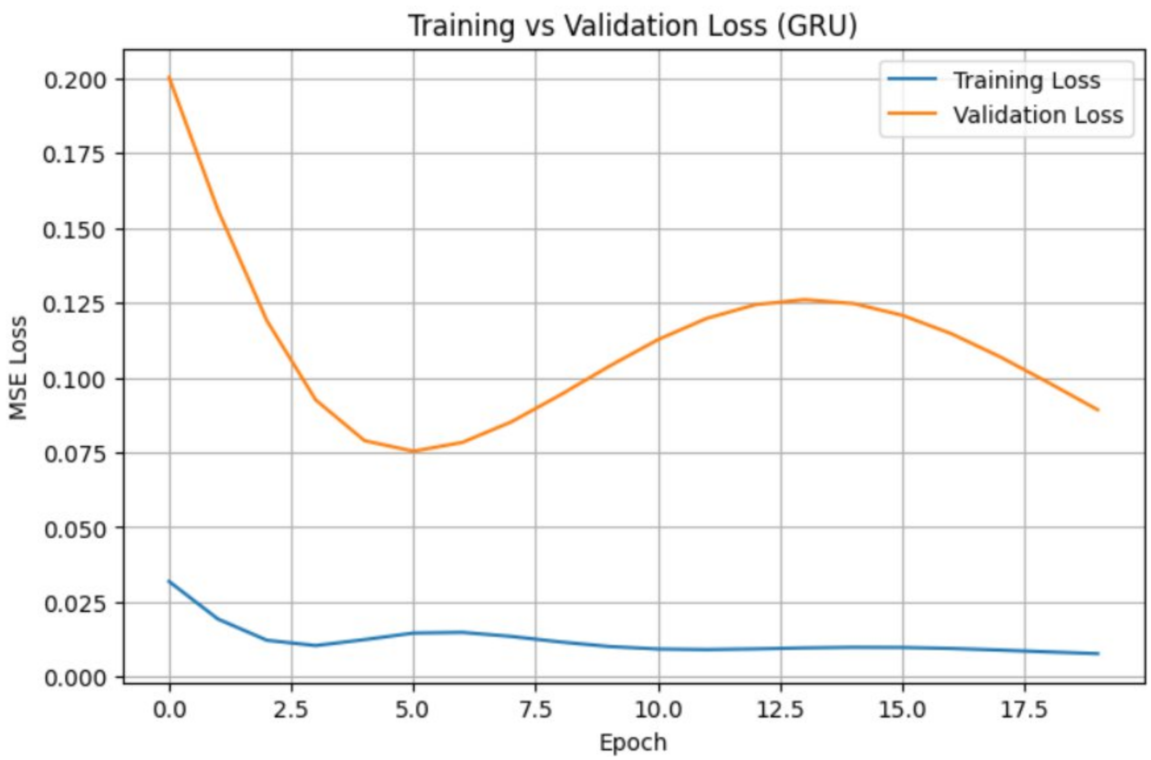
2. GRU Architecture and Training Details

A GRU neural network was implemented using PyTorch. The model consists of two GRU layers with a hidden size of 64 units, followed by a fully connected linear layer that outputs the predicted stock price.

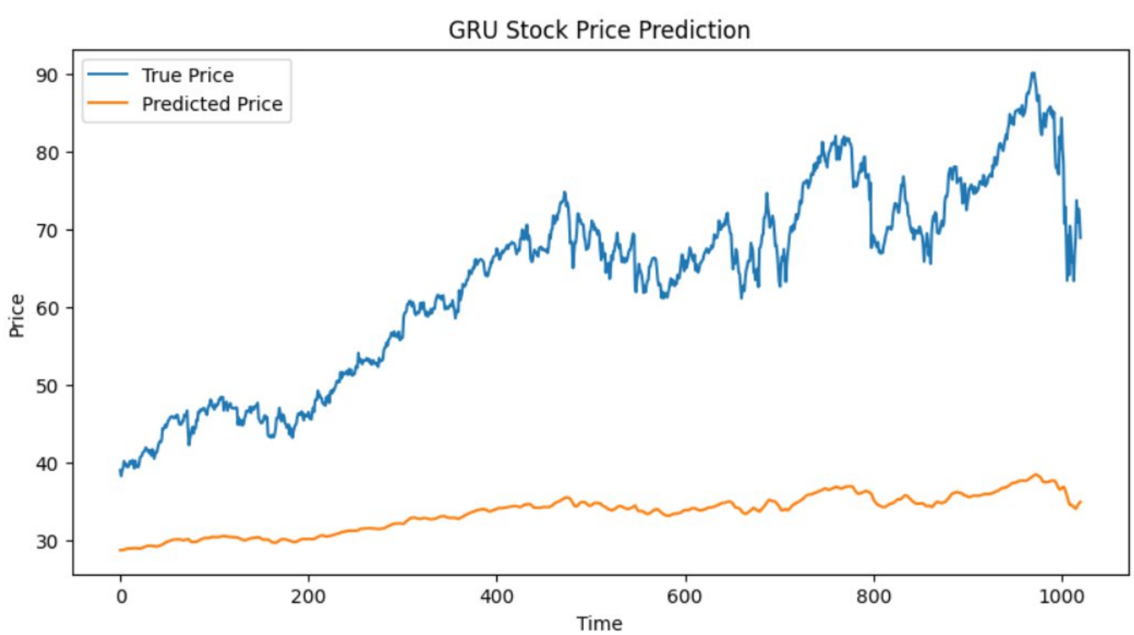
The network was trained using the MSE loss function and the Adam optimizer with a learning rate of 0.001. Training was performed for multiple epochs, while monitoring validation loss to detect overfitting.

3. Results and Analysis

The model converges rapidly, with validation loss reaching its minimum in the early epochs. Subsequent training results in minor fluctuations in validation loss, indicating stable learning with limited overfitting.



Model performance was evaluated using RMSE on the validation set. The final RMSE achieved was approximately 31.65, measured on the original price scale. Prediction plots comparing true and predicted prices show that while the model captures short-term trends, it underestimates absolute price levels.



4. Conclusions and Insights

This project demonstrates the application of GRU-based recurrent neural networks for time series prediction. While the model successfully learns temporal dependencies, stock price volatility and non-stationarity present significant challenges.

Potential improvements include predicting returns instead of raw prices, incorporating additional features such as trading volume, and experimenting with deeper architectures or regularization techniques. Future applications may include financial forecasting, risk analysis, and anomaly detection in time series data.