VIVEK DEEPAK SHETH 0890993 Project1

Stock Price Prediction Using LSTM, GRU, Fully Connected Neural Networks and GRU built from scratch.

This project implements and compares three different neural network models—LSTM, GRU, and Fully Connected Neural Network—for stock price prediction using historical data. The dataset consists of daily closing prices for the Meta stock (META) from October 7, 2019, to the present date.

Each model is trained for **100 epochs** and evaluated using **Mean Squared Error (MSE)** and **Root Mean Squared Error (RMSE)**.

Dataset

The dataset used for this project consists of historical daily **closing prices** of Meta stock (META) from **October 7, 2019**, to the **present date**. The data is obtained using the **Yahoo Finance API** via the `yfinance` library.

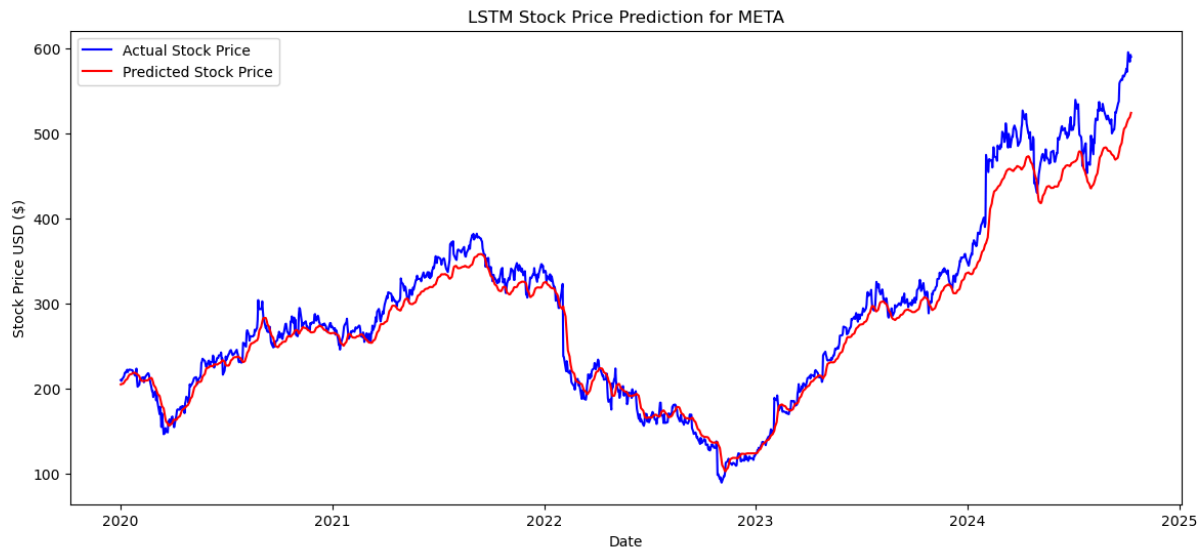
- `Close`: The closing stock price of the day.

Models

1. LSTM Model

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) capable of learning long-term dependencies. It is widely used in time-series forecasting tasks like stock price prediction.

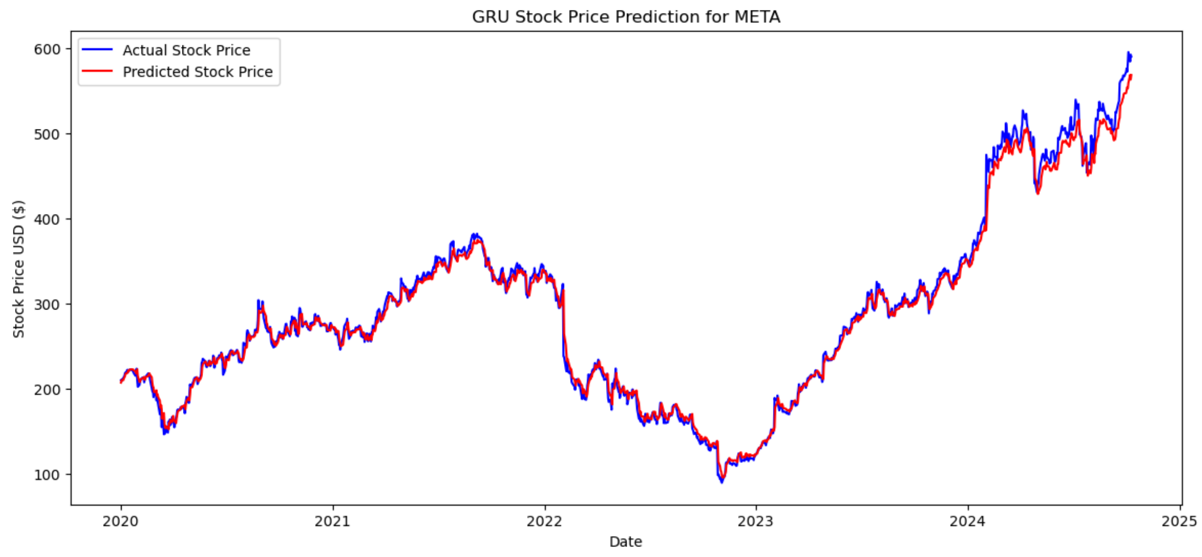
- **Architecture**:
  - Two LSTM layers with `100` units each.
  - Dropout layers with a dropout rate of `0.9` to prevent overfitting.
  - One dense layer with 1 unit to output the predicted stock price.
- **Epochs**: 100
- **Optimizer**: Adam
- **Loss Function**: Mean Squared Error
- **LSTM Model MSE**: 515.7646769846609
- **LSTM Model RMSE**: 22.7104530334527



2. GRU Model

GRU (Gated Recurrent Unit) is another variant of RNN that simplifies the LSTM model by combining the forget and input gates into a single update gate. GRUs are faster to train and can achieve similar performance to LSTM in time-series tasks.

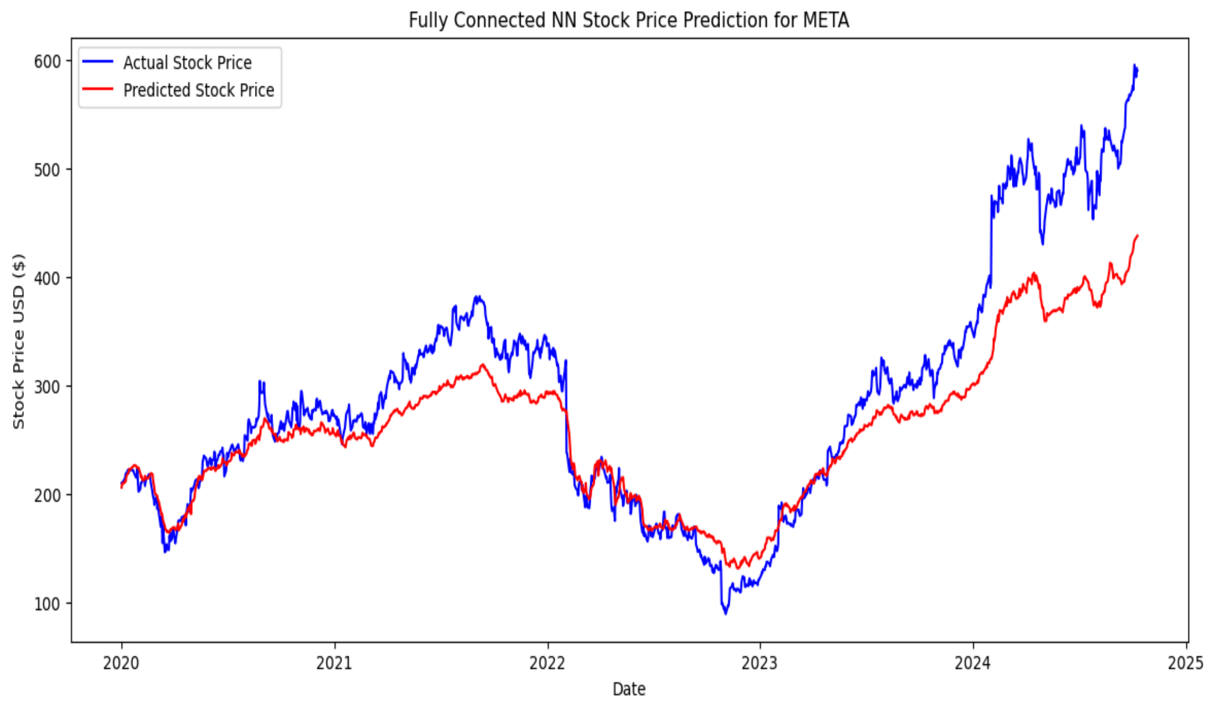
- **Architecture**:
    - Two GRU layers with `100` units each.
    - Dropout layers with a dropout rate of `0.6`.
    - One dense layer with 1 unit to output the predicted stock price.
- **Epochs**: 100
- **Optimizer**: Adam
- **Loss Function**: Mean Squared Error
- **GRU Model MSE:** 104.05016666396428
- **GRU Model RMSE:** 10.200498353706267



3. Fully Connected Neural Network

The **Fully Connected Neural Network** (FCNN) is a simpler feed-forward neural network with multiple dense layers. Although it does not handle time dependencies like LSTM or GRU, it can still capture some patterns in the data when provided with sufficient history.

- **Architecture**:
  - Two dense layers with `100` units each.
  - Dropout layers with a dropout rate of `0.2`.
  - One dense layer with 1 unit to output the predicted stock price.
- **Epochs**: 100
- **Optimizer**: Adam
- **Loss Function**: Mean Squared Error
- **Fully Connected NN Model MSE:** 2682.6717937331373
- **Fully Connected NN Model RMSE:** 51.794515093136425



4. GRU from Scratch

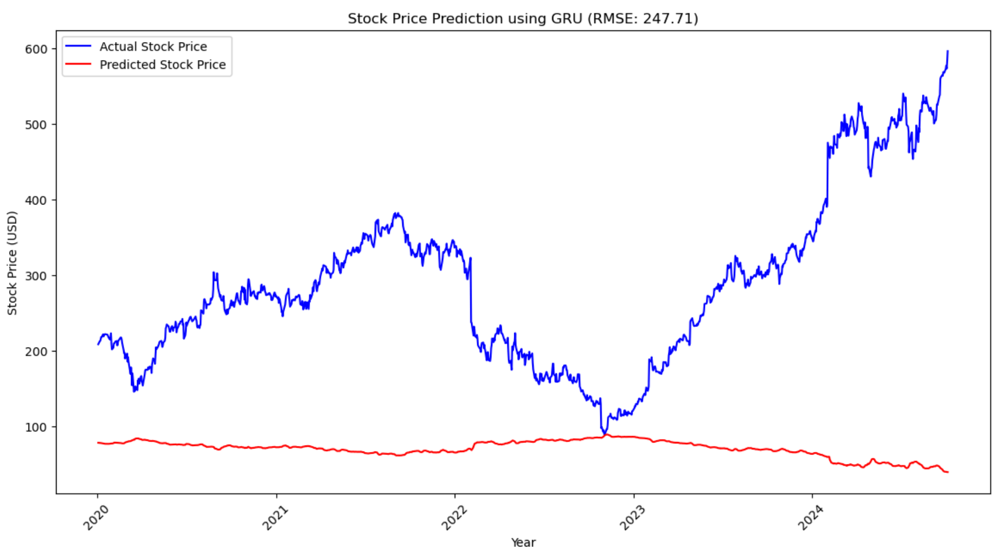
This project implements a GRU (Gated Recurrent Unit) model from scratch, without using high-level deep learning libraries like TensorFlow or PyTorch. The model is manually built using Python and `numpy` for numerical operations. Here's a brief overview of the process:

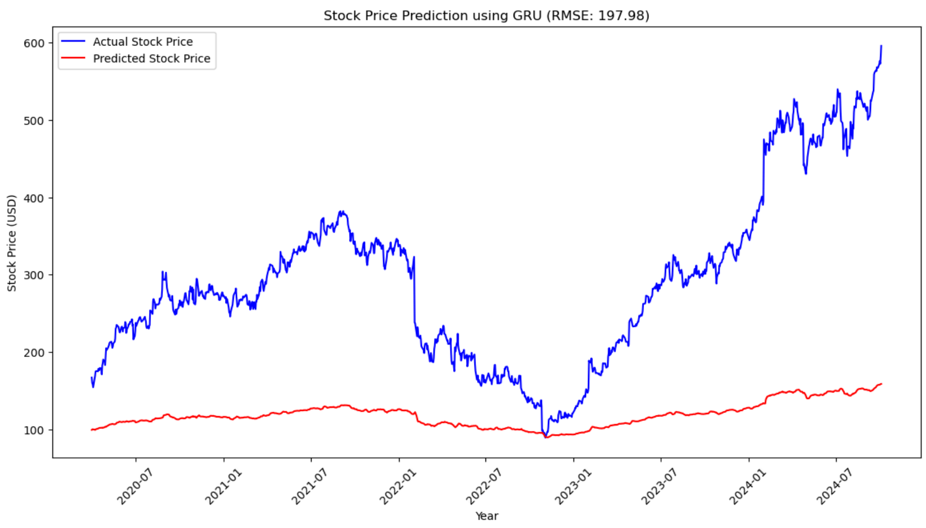
1. **GRU Architecture**: The model includes an **update gate** and **reset gate** to control the flow of information. The forward pass computes the hidden state h_t based on the input at each time step using:
   - z_t: Update gate
   - r_t: Reset gate
   - h_hat: Candidate state
   - h_t = (1 - z_t) * h_prev + z_t * h_hat
2. **Xavier Initialization**: Weights are initialized using **Xavier initialization** to improve convergence.
3. **Adam Optimizer**: The **Adam optimizer** is used for parameter updates, along with **gradient clipping** to prevent exploding/vanishing gradients.
4. **Loss Calculation**: The **Mean Squared Error (MSE)** is calculated as the loss function, comparing predicted stock prices to actual prices.
5. **Epoch Timing**: The `time` module tracks the duration of each epoch during training, providing insight into the time complexity.
6. **RMSE Evaluation**: After training, the model's accuracy is evaluated using **Root Mean Squared Error (RMSE)** to assess how well the predicted stock prices match the actual values.
7. **Plotting Results**: The predicted and actual stock prices are plotted using `matplotlib` for visual comparison.

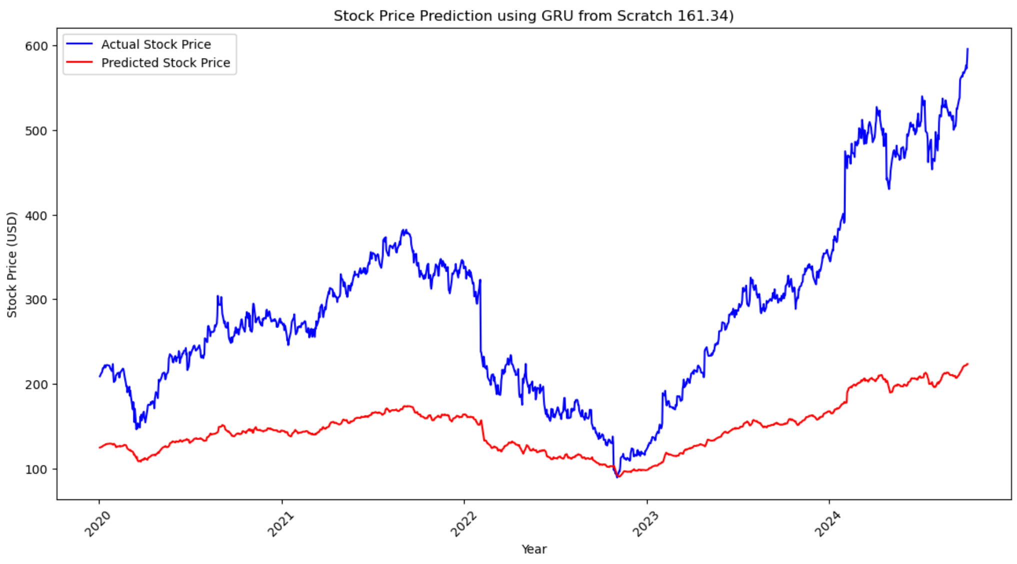
Hyperparameters

- **Learning Rate**: 0.01
- **Hidden Size**: 50
- **Epochs**: 100
- **Time Step**: 120

VIVEK DEEPAK SHETH 0890993 Project1


Stock Price Prediction using GRU (RMSE: 247.71)
Actual Stock Price
Predicted Stock Price
Stock Price (USD)
Year


Stock Price Prediction using GRU (RMSE: 197.98)
Actual Stock Price
Predicted Stock Price
Stock Price (USD)
Year


Stock Price Prediction using GRU from Scratch 161.34)
Actual Stock Price
Predicted Stock Price
Stock Price (USD)
Year