CSCI 6366

Final Project Presentation

Predicting Stock Prices with LSTM

By:
Kishan Ramesh
G44387483

Abstract

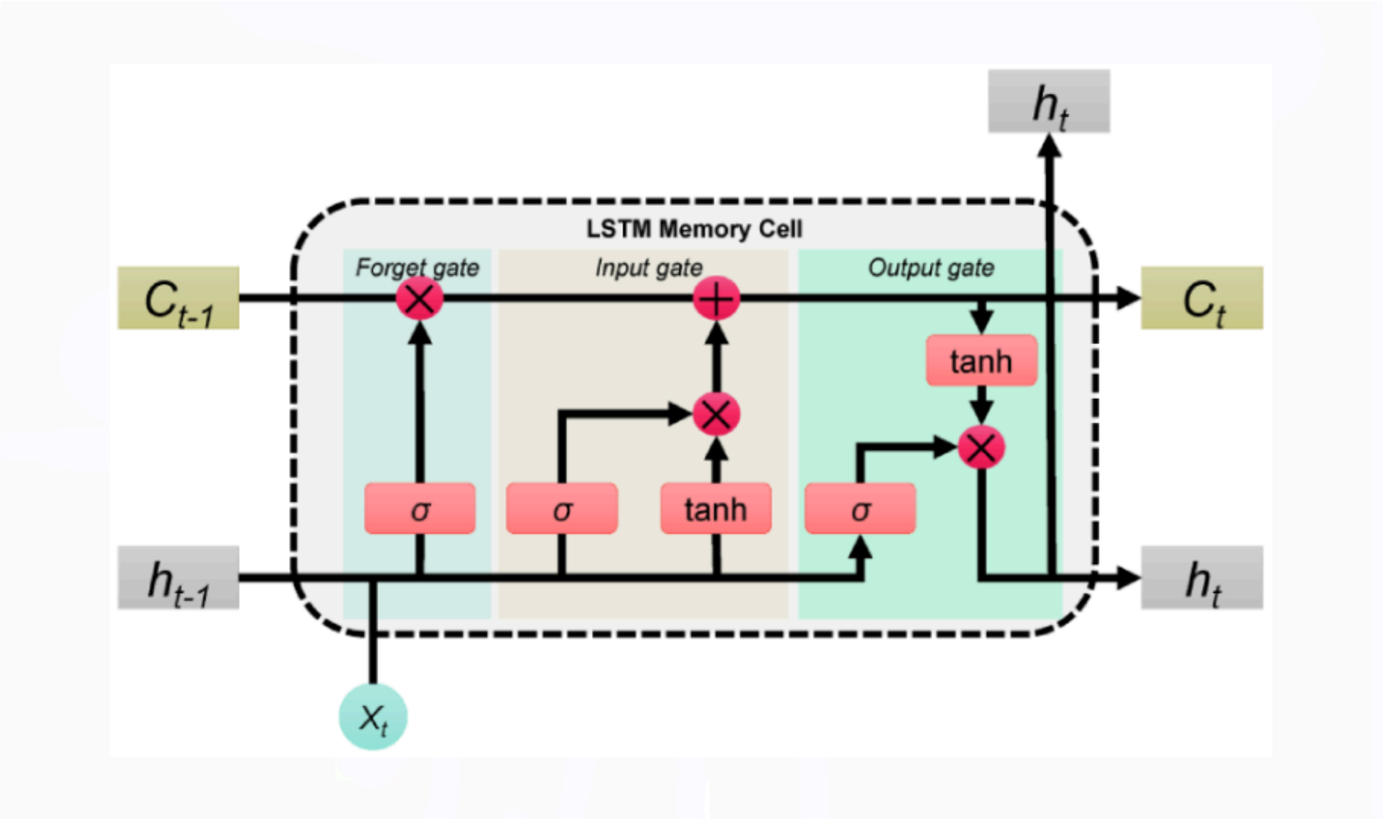
Develop an LSTM model from scratch using standard Python libraries to predict stock market closing prices for major companies, leveraging 10 years of historical data.

Introduction

- **Long Short-Term Memory (LSTM)** is a type of recurrent neural network (RNN) designed to model and predict sequences of data.
- LSTMs are equipped with a specialized architecture that addresses the problem of vanishing and exploding gradients, making them highly effective for learning long-term dependencies.
- LSTMs are widely used in applications such as time-series forecasting, natural language processing, and speech recognition, making them a powerful tool for tasks involving sequential data.

LSTM cell

An LSTM cell is composed of key components that manage the flow of information through three gates and a cell state. These components work together to learn long-term dependencies in sequential data.



- **Forget gate :**The forget gate determines what information to discard from the previous cell state.
- **Input gate :** The input gate decides what new information to store in the cell state. It consists of two steps:
    - Determine which values to update.
    - Generate candidate values to add to the cell state.
- **Cell state:** The cell state is updated by combining the scaled previous state and the new candidate information.
- **Output gate:** The output gate determines the next hidden state, which serves as output and influences future steps.

```python
def forward(self, x, h_prev, c_prev):
    combined = np.vstack((h_prev, x))  # Concatenate input and previous hidden state
    f = self.sigmoid(np.dot(self.Wf, combined) + self.bf)
    i = self.sigmoid(np.dot(self.Wi, combined) + self.bi)
    c_hat = np.tanh(np.dot(self.Wc, combined) + self.bc)
    c = f * c_prev + i * c_hat
    o = self.sigmoid(np.dot(self.Wo, combined) + self.bo)
    h = o * np.tanh(c)
    return h, c
```

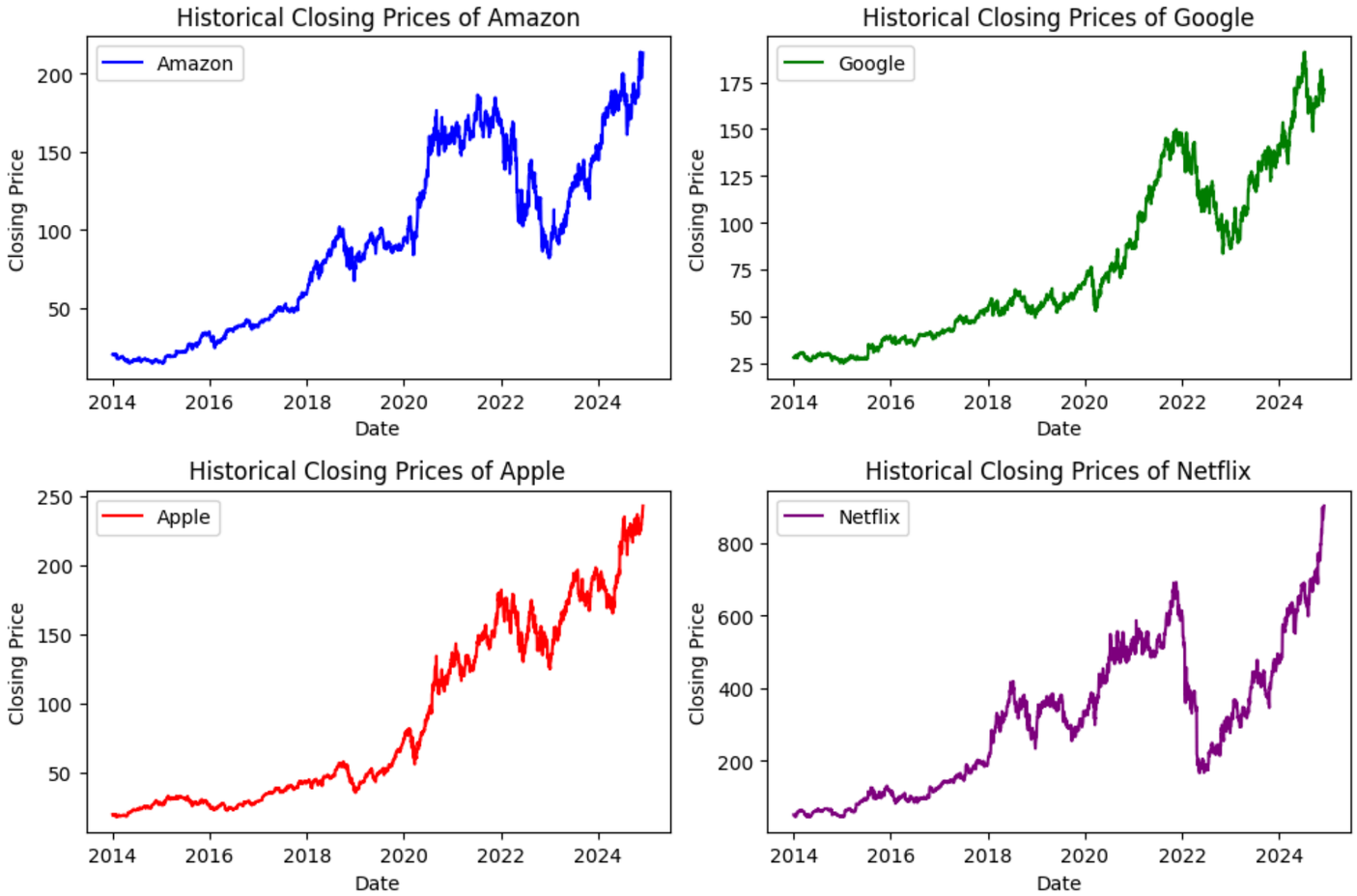
The code snippet demonstrates the implementation of all LSTM cell components within the forward propagation function of the LSTM cell class

About the Dataset

The dataset for this project consists of stock market closing prices sourced from Yahoo Finance, extracted using the Python yfinance library. It includes 10 years of historical closing prices for five major tech companies—Google, Microsoft, Apple, Netflix, and Amazon—up to the present day. The dataset is refreshed with the latest closing prices each time the notebook is run.

Ticker	Date	AMZN	GOOGL	AAPL	NFLX	MSFT
2747	2024-12-02	210.710007	171.490005	239.589996	897.739990	430.980011
2748	2024-12-03	213.440002	171.339996	242.649994	902.169983	431.200012
2749	2024-12-04	218.160004	174.369995	243.009995	911.059998	437.420013
2750	2024-12-05	220.550003	172.639999	243.039993	917.869995	442.619995
2751	2024-12-06	227.029999	174.710007	242.839996	934.739990	443.570007

Visualization



Data PreProcessing

- In the data preprocessing step, Min-Max Scaling is applied to normalize the numerical data, ensuring it is transformed into a range between 0 and 1 while maintaining the original distribution characteristics.
- Next step involves creating datasets for the LSTM model by generating input-output pairs, where each input sequence consists of a specified number of past time steps (e.g., 60), and the corresponding output is the value at the next time step.
- The time step window captures temporal dependencies and trends in the data, enabling the model to learn patterns and relationships in historical stock closing prices, which are crucial for accurate predictions.

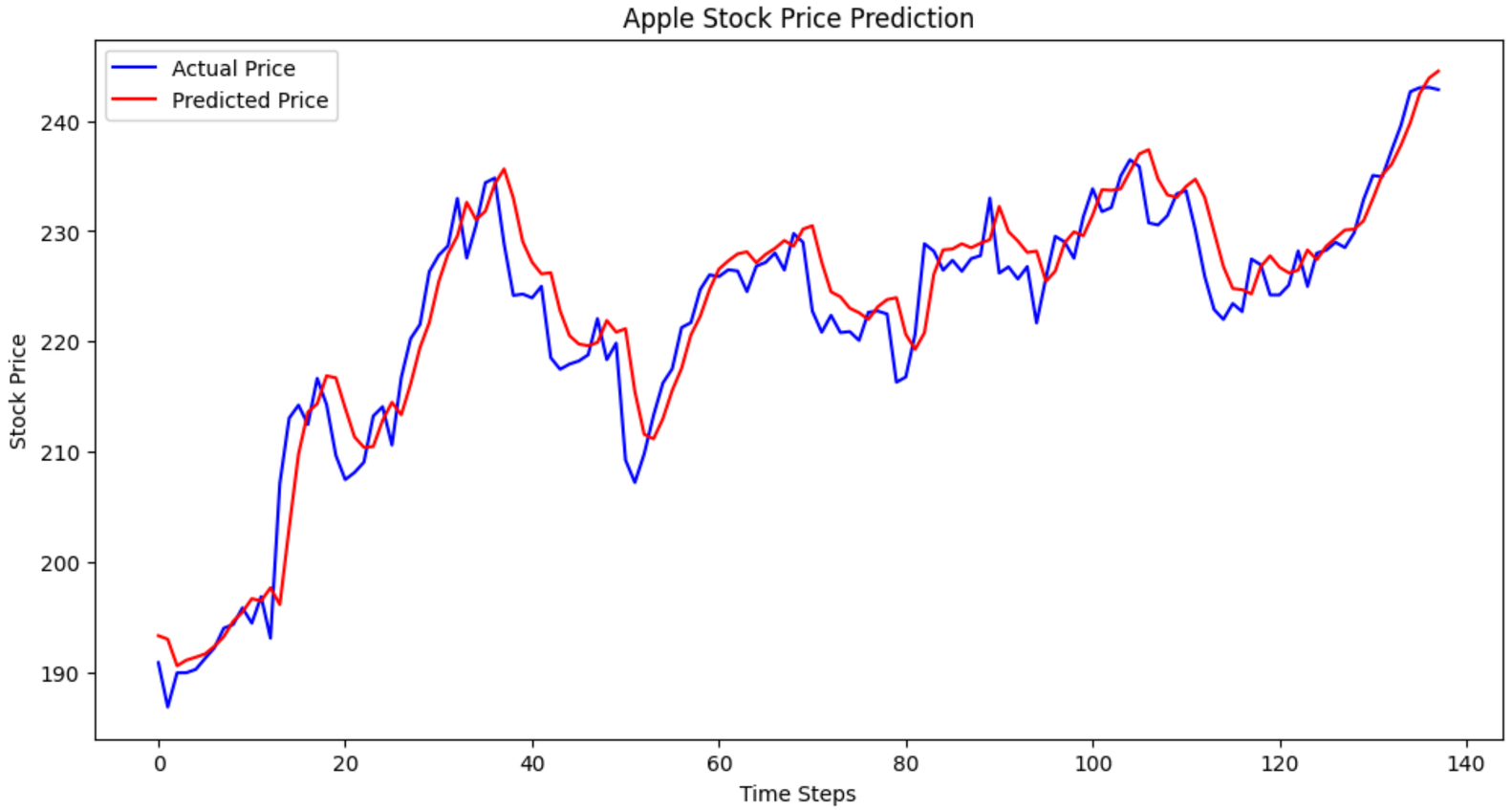
Model Training

The LSTM model training involves:

- **Forward Propagation**: The input sequence is passed through the LSTM cell, updating the hidden and cell states for each time step, and producing the final output prediction.
- **Loss Calculation**: The model computes the loss by comparing the predicted output with the true labels using mean squared error (MSE), quantifying the error.
- **Backpropagation Through Time (BPTT)**: Gradients of the loss with respect to the model parameters are calculated by unrolling the LSTM over time and applying the chain rule across all time steps.

- **Gradient Clipping**: The computed gradients are clipped to a predefined range (e.g., -1.0 to 1.0) to prevent exploding gradients and ensure stable training.
- **Adam Optimizer**: The weights are updated using the Adam optimization algorithm, which combines momentum and adaptive learning rates for efficient convergence.
- **Epochs Iteration**: The training process is repeated for a predefined number of epochs, with the loss tracked and minimized progressively.

Results



The graph illustrates the predicted closing prices of Apple stock, evaluated on the test dataset after training the model. The trendlines of the actual and predicted prices are closely aligned but not identical.

```
The forecasted closing price of APPLE is :  244.6453830479852
```

The image displays the predicted closing price of Apple stock for the following day.

```
Mean Absolute Error (MAE): 12.772188148873333
Mean Squared Error (MSE): 293.70565989942105
```

The image presents the loss metrics computed from the model's predictions on the test dataset.

Limitations

- A slight right deviation in the predicted price trendline compared to the original closing prices, along with a slightly higher error, has been observed.
- LSTMs often face challenges in accurately capturing long-term dependencies, particularly in noisy and highly non-linear datasets like stock prices.
- The inherent volatility of stock market data makes it difficult for LSTMs to extract meaningful patterns without being affected by noise.
- While LSTMs are well-suited for sequential data with stable patterns, they may struggle to model the long-term dependencies crucial for financial data.

Conclusion

- In conclusion, the primary objective of the project—to build an LSTM model from scratch and utilize it for predicting the stock market closing prices of tech companies—has been successfully achieved.
- Despite minor deviations in the predictions, the model demonstrates satisfactory performance on the given stock market dataset.
- However, the limitations observed can be addressed by incorporating pre-trained models, Hybrid Attention-LSTM models, or Vanilla Transformer models, which offer improved handling of long-term dependencies and a deeper understanding of complex patterns in financial data.

Thank you

Time for Q&A