CS-GY 6953 Final Project
Forecasting of FOREX Price Distribution with LSTM Neural Network

Nikhil Amin, Ping Chang, Niko Ganev
https://github.com/nikoaganev/deepLearning6953

Introduction

The Foreign Exchange Market is a financial market where currencies are bought and sold simultaneously. In this market, currencies are denominated in pairs, for example the price of USD in EUR is the EUR/USD pair, and price of USD in Japanese Yen is JPY/USD.

Using data for defined time intervals, our team aims to build a model that predicts the direction of the "closing price" for the next time period for a given pair of currencies with an accuracy superior to 50%, as 50% is the equivalent of choosing randomly or flipping a coin.

Our team decided to use Long-Short-Term Memory (LSTM) network with dropout layers. We focused on testing for the accuracy of direction prediction when the model predicts a small price move vs. when it predicts a much bigger change in price. This decision was due to a lower accuracy when the model predicted larger price changes as these type of price movements are typically caused by a change in macro-economic factors or the release of important news which we are not taking into consideration in this experiment to avoid over-complicating the project.

The results show that it is possible to predict the direction of currency price movement over a duration of the next 15 minutes with an accuracy of 65%. Although to get an accuracy of over 60%, we had to feed in relatively small data, the model still shows its capability of predicting better than relying on a random outcome.

Methodology

The model used for this project is a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). In an LSTM, there are three types of nodes: input, forget, and output (Hochreiter, 1997). These nodes decide how each input would affect the model when creating the memory component. LSTMs are capable of learning long-term dependencies, especially in sequence prediction problems, as well as having feedback connections that allow for processing of entire sequences of data (Lawi, 2021) The reason our team picked this model is because it performs well with sequential data, such as time series, which is the type of data used when forecasting currency or stock prices.

In an RNN, the information goes through a cycle (Lima, 2021). The model considers the current input in addition to what it has learned from the previously received inputs before making a decision. Figure 1 shows what the structure of a recurrent neural network looks like.

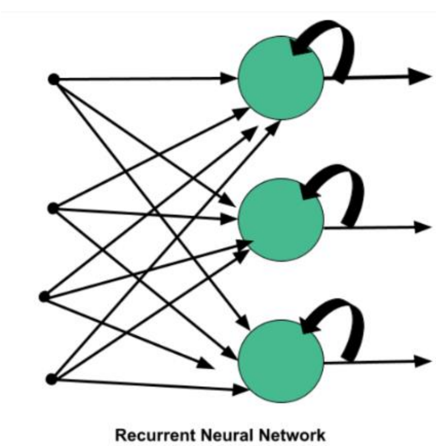

Figure 1: Recurrent Neural Network

Our final model architecture was achieved by alternating LSTM layers with dropout layers. These layers are used to randomly drop some of the features before injecting into the next layer to capture specific patterns and improve accuracy. Our team also experimented by varying the number of these dropout layers before settling for the final architecture which consisted of four LSTM layers with four dropout layers.

The initial input is a sequence of currency prices (Qu, 2019) from a set number of previous closing prices called the “sliding window” (Yildirim, 2021). We experimented with windows of different sizes. For example, using the previous five 75 minutes price fluctuation to predict the next 15 minutes, or the use of the previous several hours of price fluctuations to predict the next 15 minutes. The final model uses a sliding window of size 5. As we are working with 15-minute price movements this is the equivalent of using the previous 75 minutes to predict the next 15 mins [Figure 2].

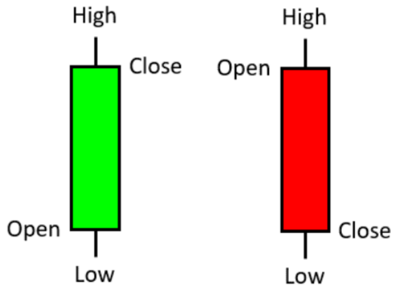

Figure 2: Candlesticks Depicting Price Movement Over a Period of Time

The training data was scaled using a Min-Max scaler as our model predicts a price rather than a direction. This was also proven to increase performance. Based on the predicted price, we infer the predicted direction of price movement by comparing it to the previous price prediction to determine if the model thinks the price will go up or down.

The mean square error loss function was used to train our model as it attempts to predict a currency price rather than a classification. Using this loss function, we calculate how far the prediction was from the actual price before using back-propagation.

Figure 3: Sliding Window Visualization and Example of Small and Large Price Moves

We attempted approaching the problem directly as a classification problem with cross-entropy loss, but the accuracy remained low despite experimenting with various architectures and parameters. Predicting the price and inferring the direction from the price prediction achieved significantly better results.

The performance of our model was evaluated based on the number of times the direction of the price-move was correctly predicted.

In the proposal for this project, our team mentioned that macro-economic data and NLP monitoring news will not be incorporated because of complexity. These factors do cause quick adjustments to the price of currencies, which results in sharp moves. These sharp movements can be easily spotted on a chart because the price, in each timeframe, goes up and down by a much larger amount than usual. This is shown in Figure 3. As a result, when monitoring our results, we focused on the accuracy of direction prediction when the model predicts a small price move rather than when it predicts a much bigger change in price.

The architecture of our final model is presented in Figure 4.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_96 (LSTM)               (None, 5, 100)            42400

dropout_96 (Dropout)         (None, 5, 100)            0

lstm_97 (LSTM)               (None, 5, 5)              2120

dropout_97 (Dropout)         (None, 5, 5)              0

lstm_98 (LSTM)               (None, 5, 120)            60480

dropout_98 (Dropout)         (None, 5, 120)            0

lstm_99 (LSTM)               (None, 180)               216720

dropout_99 (Dropout)         (None, 180)               0

dense_24 (Dense)             (None, 1)                 181

=================================================================
Total params: 321,901
Trainable params: 321,901
Non-trainable params: 0
```

Figure 4: Final Model Structure

Results

The main hyperparameters we fine-tuned are the:

- The size of the sliding window
- The number of epochs
- The learning rate
- The batch size
- The dataset size

Below are the charts representing the results for each one of these parameters.

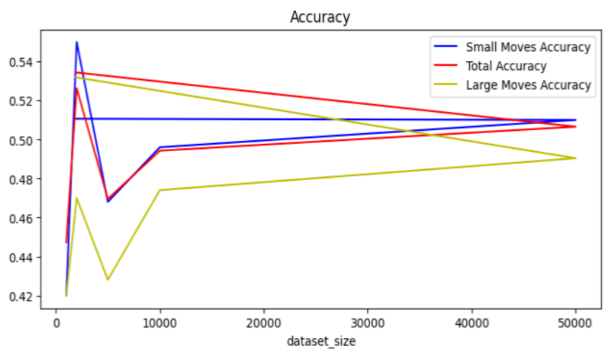

Figure 5: Accuracy of Model for Dataset Sized from 2,000 to 50,000 Datapoints.

The **size of the data set** is likely the most important parameter. Surprisingly, in this case more data means smaller accuracy. This make sense in our case as we are trying to predict what the stock price is going to do over a very short period of time and for that reason looking at price behavior for a year or a month ago does not provide the model with the most insight. It seems like focusing on the most recent price sequence is what allows us to break out of the 45% to the 55% range of accuracy and even go over 60%. Taking 1500 to 2000 data points which represents about 15 to 20 days of trading is optimal.

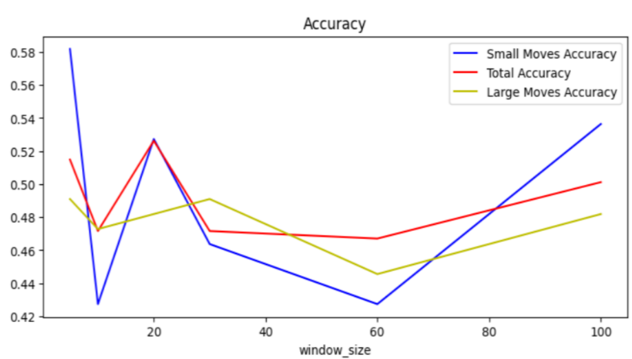

Figure 6: Accuracy of Model for Sliding Window Sizes Between 5 and 60 Time Periods.

Window Size represents how much of the previous price fluctuation we take into consideration to make a prediction on the direction of the price over the next 15 minutes (Qi, 2021). Based on the results above, the window size is an important factor, and it seems like shorter windows translate into higher accuracies. The final model uses a window of 5 candles which is 75mins to predict the next 15mins.

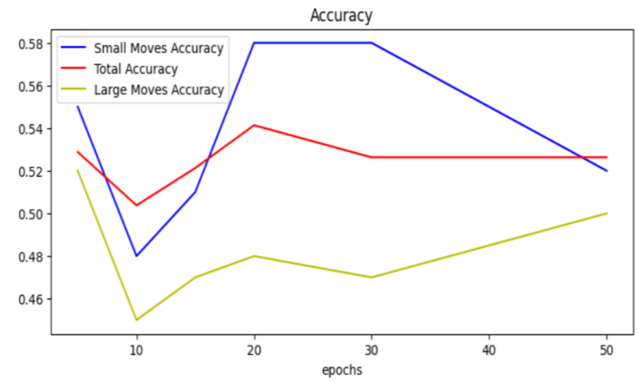

Figure 7: Accuracy of Model for Number of Training Epochs Between 5 and 50

The **number of epochs** the model is trained has some effect on the accuracy but going over 20 epochs for a small dataset seem to quickly result in over fitting and a decrease of accuracy on the test data.

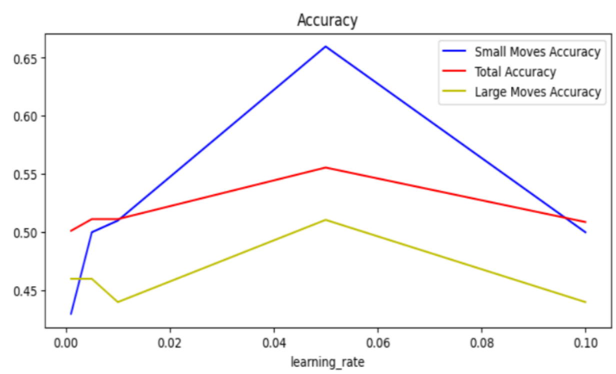

Figure 8: Accuracy of Model for Learning Rates Between 0.001 and 0.1

The **learning rate** affect the results and is a key metric in improving performance, based on the chart above, a learning rate around 0.05 is optimal. Smaller learning rates than 0.05 cause overfitting and a risk of vanishing gradients.

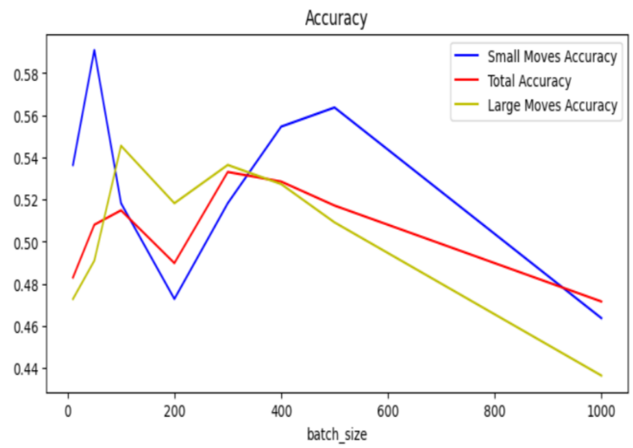

Figure 9: Accuracy of Model of Training Batch Sizes Between 10 and 1000

The **batch size** used for each epoch also has a major effect on the accuracy and batches of around 500 datapoints seem to be optimal. Using a sliding window of the 5 previous timeframes is actually equivalent to batches of 2500 datapoints divided in sequences of 5 sequential prices per input.

Conclusion

In conclusion, it is possible to predict the direction of currency price movements over the next 15 minutes with an accuracy superior to 60%. Our final model has an accuracy of 65%!

Furthermore, to achieve this type of accuracy without considering macro-economic data and news a few aspects need to be considered.

The model needs to be fed relatively small data because patterns that might have led to an increase in price in the past might in more recent times mean a decrease is coming and provide conflicting information to the model. The last few weeks of price behavior is sufficient. For the same reason, the model will not stay accurate long and so it must be retrained very often. If used in production, then maybe retraining it at least daily will be necessary.

Also, when there are changes in the macro-economic environment or major news releases that have a significant effect on the price, the model will not perform well and a trader using this type of model would have to wait for the news be fully reflected in the price before resuming the use of the model to predict the direction of the next trade.

References

Accurately Forecasting Stock Prices using LSTM and GRU Neural Networks
Lawi, A.; Kurnia E. 2021. A Deep Learning Approach for Forecasting Stock Price Time-Series Data in Groups. Lambert Academic Publishing

Application of LSTM Neural Network in Forecasting Foreign Exchange Price
Qu, Y.; Zhao X. 2019. LSTM Neural Network Model and Its Characteristics.

A Bayesian Regularized Artificial Neural Network for Stick Market Forecasting
Ticknor, J. 2013. Bayesian Regularization of Neural Networks.

Bitcoin Price Prediction Using Recurrent Neural Networks and LSTM
Lima, A. 2021. Long Short-Term Memory (LSTM).

Event-Driven LSTM For Fortex Price Prediction
Qi, L.; Khushi, M.; Poon J. 2021. Feature and Training Data Selection.

Forecasting Directional Movement of Forex Data Using LSTM with Technical and Macroeconomic Indicators
Yildirim, D.; Toroslu, I. 2021. Macroeconomic and Technical LSTM Model Results.

Neural Computation
Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory 9(8), 1735–1780 (1997).