Directional Tick Forecasting with Multivariate Time-Series Data

Adam Rolander

Background

Mentor

Brent Dornier

- Vice President of Trading at Strix Leviathan
- Introduced through Mrs. Dornier

Time Frame

Summer 2023-Present

- Had video calls & began studying background info last summer
- Spent ~40 hours on project

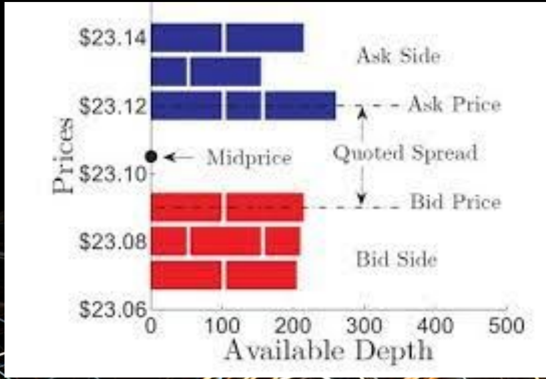
Initial Goals

CS & Math

- Shadow a career using both of these fields
- Learn about microeconomics and machine learning applications

High Frequency Trading

- Sophisticated market participants can capitalize on informational advantages, place limit orders on the Limit Order Book (LOB), and provide liquidity as market makers
    - Limit orders prioritize the price of a trade, not immediate realization
- Other market participants place market orders, cross the bid-ask spread, and pay more to execute existing limit orders immediately
    - Causes changes in price
- Orders are placed and filled electronically by trading algorithms within millisecond intervals
- LOB data is stored in databases



High Frequency Trading

- LOB data is dense and seemingly random to human eyes

    ** Price changes are never truly random unless all available information is known by every market participant (Pareto Optimality/Market Efficiency)*

- LOB data can be used to deduce trends and predict future price movements

    - Knowing future price movements (tick direction) allows market makers to place orders strategically, minimize losses, and earn profits

So…

How can we find trends in Limit Order Book data in order to forecast price tick direction?

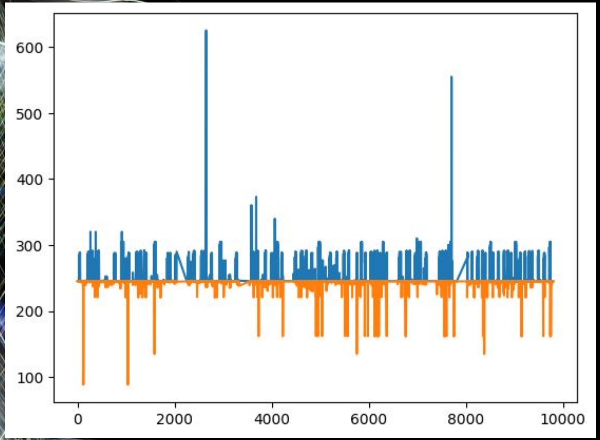
BID	1691509347145	1691509347026	8680918711	8680918712	244.90000000	531.43500000	DELTA
BID	1691509347145	1691509347026	8680918711	8680918712	244.70000000	671.47300000	DELTA
BID	1691509347301	1691509347126	8680918713	8680918713	244.70000000	676.37700000	DELTA
ASK	1691509347438	1691509347326	8680918714	8680918714	245.20000000	552.79800000	DELTA
ASK	1691509347531	1691509347426	8680918715	8680918715	245.20000000	552.68300000	DELTA
ASK	1691509347630	1691509347526	8680918716	8680918716	245.20000000	552.59900000	DELTA
ASK	1691509347731	1691509347626	8680918717	8680918717	245.50000000	430.30000000	DELTA
ASK	1691509347830	1691509347726	8680918718	8680918719	245.20000000	555.90600000	DELTA
ASK	1691509347830	1691509347726	8680918718	8680918719	245.50000000	428.02900000	DELTA
ASK	1691509348031	1691509347926	8680918720	8680918720	245.20000000	552.59900000	DELTA
ASK	1691509348132	1691509348026	8680918721	8680918722	245.50000000	430.30000000	DELTA
BID	1691509348132	1691509348026	8680918721	8680918722	244.70000000	671.47300000	DELTA
ASK	1691509348233	1691509348126	8680918723	8680918724	245.20000000	552.35900000	DELTA

L S T M

Long Short-Term Memory
Neural Network

Data Preprocessing

- Mentor provided a LOB dataset for the Binance (BNB) token
    - 5th largest cryptocurrency
    - ~ 10,000 data points
- Initial step was to clean and separate the dataset into usable parts
- Separated into Bid/Ask dataframes, dropped several initial fields, left with **time, price, and volume** (size) as input factors

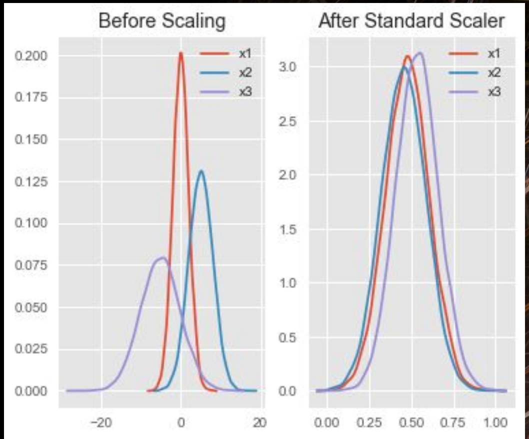


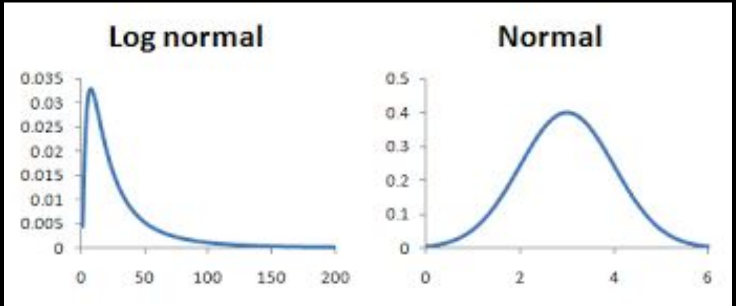
	Side	Received Time	API Time	Price	Size	Style
0	ASK	1691509347031	1691509346926	245.4	494.380	DELTA
6	ASK	1691509347438	1691509347326	245.2	552.798	DELTA
7	ASK	1691509347531	1691509347426	245.2	552.683	DELTA
8	ASK	1691509347630	1691509347526	245.2	552.599	DELTA
9	ASK	1691509347731	1691509347626	245.5	430.300	DELTA
...	...	...	...	...	...	...
9784	ASK	1691510081208	1691510081103	245.0	801.692	DELTA
9786	ASK	1691510081408	1691510081304	245.0	799.692	DELTA
9794	ASK	1691510082711	1691510082604	245.0	799.274	DELTA
9796	ASK	1691510082909	1691510082804	245.1	401.131	DELTA
9797	ASK	1691510083012	1691510082904	245.0	799.691	DELTA

	Side	Received Time	API Time	Price	Size	Style
1	BID	1691509347031	1691509346926	245.1	953.436	DELTA
2	BID	1691509347031	1691509346926	245.0	776.658	DELTA
3	BID	1691509347145	1691509347026	244.9	531.435	DELTA
4	BID	1691509347145	1691509347026	244.7	671.473	DELTA
5	BID	1691509347301	1691509347126	244.7	676.377	DELTA
...	...	...	...	...	...	...
9791	BID	1691510082009	1691510081904	244.9	76.933	DELTA
9792	BID	1691510082410	1691510082304	244.9	76.398	DELTA
9793	BID	1691510082509	1691510082404	244.9	75.864	DELTA
9795	BID	1691510082810	1691510082704	244.9	75.597	DELTA
9798	BID	1691510083012	1691510082904	244.7	1013.829	DELTA

Data Scaling

- Data had to be re-scaled before training the neural network
- Initially used Standard Scaler (z-score element-wise scaling)
    - Was unsuccessful because original data was log-normally distributed
- Used logarithmic scaling to achieve batch normalization





$$Z = \frac{x - \mu}{\sigma}$$

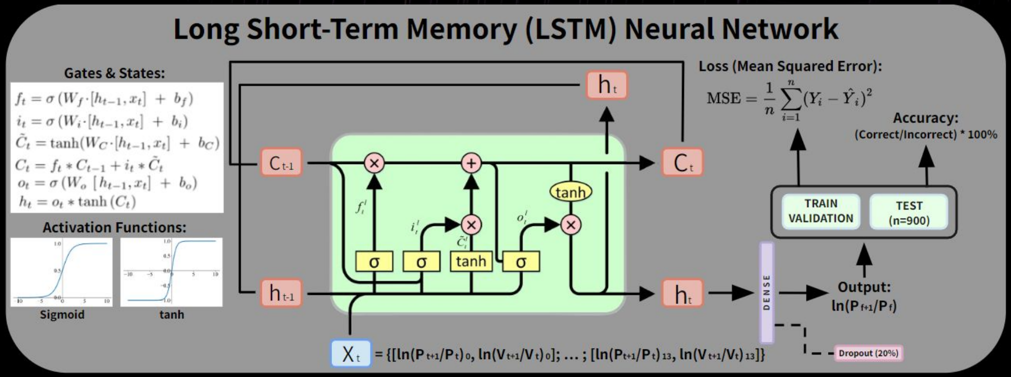
$$L = ln(P_1/P_0)$$

LSTM Implementation

- LSTM is a type of Recurrent Neural Network (RNN)
  - Useful for time-series forecasting & learning long-term dependencies
  - LSTM is unique by avoiding vanishing/exploding gradients
- Built 16 iterations of LSTM over 8 different trials
  - 2 per trial, 1 for Bid & 1 for Ask
  - Each had 29,601 parameters & trained over 10 epochs
- Trained on ~80% of initial dataframe
- Forecasted 900 values each & compared with remaining 20% of initial data

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 14, 64)	17152
lstm_1 (LSTM)	(None, 32)	12416
dropout (Dropout)	(None, 32)	0
dense (Dense)	(None, 1)	33

Total params: 29601 (115.63 KB)
Trainable params: 29601 (115.63 KB)
Non-trainable params: 0 (0.00 Byte)




```
Epoch 1/10
184/184 [==============================] - 6s 15ms/step
Epoch 2/10
184/184 [==============================] - 2s 13ms/step
Epoch 3/10
184/184 [==============================] - 2s 13ms/step
Epoch 4/10
184/184 [==============================] - 2s 13ms/step
Epoch 5/10
184/184 [==============================] - 4s 20ms/step
Epoch 6/10
184/184 [==============================] - 3s 15ms/step
Epoch 7/10
184/184 [==============================] - 2s 13ms/step
Epoch 8/10
184/184 [==============================] - 2s 13ms/step
Epoch 9/10
184/184 [==============================] - 2s 13ms/step
Epoch 10/10
184/184 [==============================] - 3s 18ms/step
```

Results

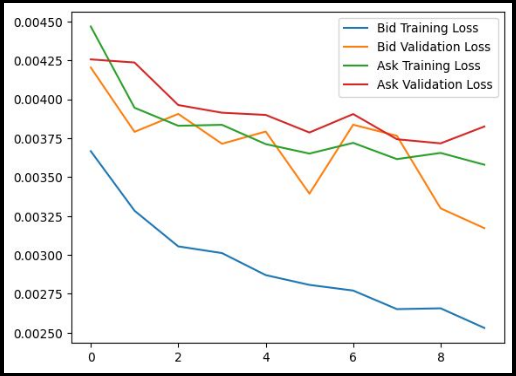
- Training and validation loss were evaluated by Mean Squared Error: $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$
    - Loss decreased over training cycles (epochs)
- Forecasted values were compared to original data
    - If corresponding values had the same sign (++ or --), tick direction was forecasted correctly
    - Calculated percent accuracy
- Best trial achieved 65.56% for Ask and 66.89% for Bid

```
# ASK TICK VALIDATION

correct = 0
incorrect = 0

for i in range(m_future):
  if((forecast[i] >= 0 and ln_price_quotient_test[i] >= 0)
  or forecast[i] <= 0 and ln_price_quotient_test[i] <= 0):
    correct += 1
  else:
    incorrect += 1

print("Correct: {}".format(correct))
print("Incorrect: {}".format(incorrect))
print("Total: {}".format(m_future))
```

```
Correct: 590
Incorrect: 310
Total: 900
```

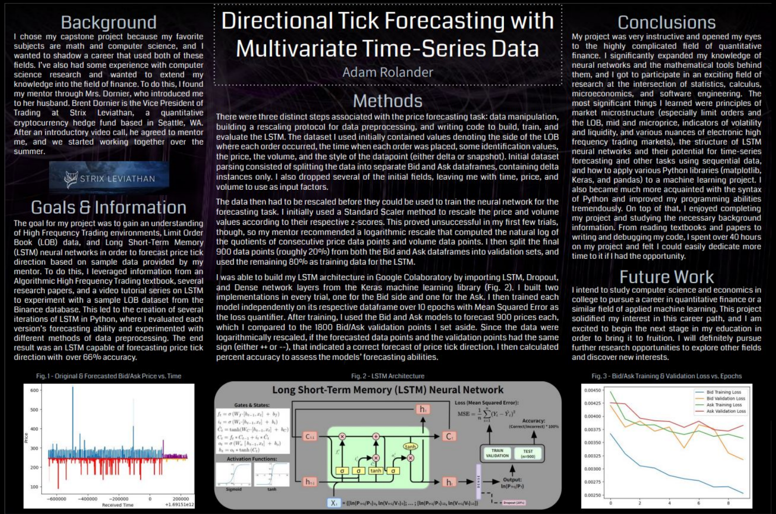
```
# BID TICK VALIDATION

correct = 0
incorrect = 0

for i in range(m_future):
  if((forecast_b[i] >= 0 and ln_price_quotient_b_test[i] >= 0)
  or forecast_b[i] <= 0 and ln_price_quotient_b_test[i] <= 0):
    correct += 1
  else:
    incorrect += 1

print("Correct: {}".format(correct))
print("Incorrect: {}".format(incorrect))
print("Total: {}".format(m_future))
```

```
Correct: 602
Incorrect: 298
Total: 900
```



Conclusions & Future Work

- Project was very instructive
- Would be interested in revisiting in the future
    - Other cryptocurrencies, different network architectures/input features
- Also helpful with my future plans
- Study CS/Econ in college
- Want to pursue a career in quantitative finance or applied machine learning



Directional Tick Forecasting with Multivariate Time-Series Data

Adam Rolander

Background

I chose my capstone project because my favorite subjects are math and computer science, and I wanted to shadow a career that used both of these fields. I've also had some experience with computer science research and wanted to extend my knowledge into the field of finance. To do this, I found my mentor through Mrs. Dornier, who introduced me to her husband. Brent Dornier is the Vice President of Trading at Strix Leviathan, a quantitative cryptocurrency hedge fund based in Seattle, WA. After an introductory video call, he agreed to mentor me, and we started working together over the summer.

STRIX LEVIATHAN

Goals & Information

The goal for my project was to gain an understanding of High Frequency Trading environments, Limit Order Book (LOB) data, and Long Short-Term Memory (LSTM) neural networks in order to forecast price tick direction based on sample data provided by my mentor. To do this, I leveraged information from an Algorithmic High Frequency Trading textbook, several research papers, and a video tutorial series on LSTM to experiment with a sample LOB dataset from the Binance database. This led to the creation of several iterations of LSTM in Python, where I evaluated each version's forecasting ability and experimented with different methods of data preprocessing. The end result was an LSTM capable of forecasting price tick direction with over 66% accuracy.

Methods

There were three distinct steps associated with the price forecasting task: data manipulation, building a rescaling protocol for data preprocessing, and writing code to build, train, and evaluate the LSTM. The dataset I used initially contained values denoting the side of the LOB where each order occurred, the time when each order was placed, some identification values, the price, the volume, and the style of the datapoint (either delta or snapshot). Initial dataset parsing consisted of splitting the data into separate Bid and Ask dataframes, containing delta instances only. I also dropped several of the initial fields, leaving me with time, price, and volume to use as input factors.

The data then had to be rescaled before they could be used to train the neural network for the forecasting task. I initially used a Standard Scaler method to rescale the price and volume values according to their respective z-scores. This proved unsuccessful in my first few trials, though, so my mentor recommended a logarithmic rescale that computed the natural log of the quotients of consecutive price data points and volume data points. I then split the final 900 data points (roughly 20%) from both the Bid and Ask dataframes into validation sets, and used the remaining 80% as training data for the LSTM.

I was able to build my LSTM architecture in Google Colaboratory by importing LSTM, Dropout, and Dense network layers from the Keras machine learning library (Fig. 2). I built two implementations in every trial, one for the Bid side and one for the Ask. I then trained each model independently on its respective dataframe over 10 epochs with Mean Squared Error as the loss quantifier. After training, I used the Bid and Ask models to forecast 900 prices each, which I compared to the 1800 Bid/Ask validation points I set aside. Since the data were logarithmically rescaled, if the forecasted data points and the validation points had the same sign (either ++ or --), that indicated a correct forecast of price tick direction. I then calculated percent accuracy to assess the models' forecasting abilities.

Conclusions

My project was very instructive and opened my eyes to the highly complicated field of quantitative finance. I significantly expanded my knowledge of neural networks and the mathematical tools behind them, and I got to participate in an exciting field of research at the intersection of statistics, calculus, microeconomics, and software engineering. The most significant things I learned were principles of market microstructure (especially limit orders and the LOB, mid and microprice, indicators of volatility and liquidity, and various nuances of electronic high frequency trading markets), the structure of LSTM neural networks and their potential for time-series forecasting and other tasks using sequential data, and how to apply various Python libraries (matplotlib, Keras, and pandas) to a machine learning project. I also became much more acquainted with the syntax of Python and improved my programming abilities tremendously. On top of that, I enjoyed completing my project and studying the necessary background information. From reading textbooks and papers to writing and debugging my code, I spent over 40 hours on my project and felt I could easily dedicate more time to it if I had the opportunity.

Future Work

I intend to study computer science and economics in college to pursue a career in quantitative finance or a similar field of applied machine learning. This project solidified my interest in this career path, and I am excited to begin the next stage in my education in order to bring it to fruition. I will definitely pursue further research opportunities to explore other fields and discover new interests.

Fig. 1 - Original & Forecasted Bid/Ask Price vs. Time

Fig. 2 - LSTM Architecture

Fig. 3 - Bid/Ask Training & Validation Loss vs. Epochs

Thank You!
Questions?