1- Does this data set even make sense? What are the limitations of this data set?

The data set would make sense if you were trying to predict second-by-second intervals in an isolated market. However, the data and its features are very limited in a practical sense -- because they show only a limited picture of the cause of the price action, there can and will be a lot of unexplained volatility in the data. From my experience working on an economics team, I have seen how interconnected markets truly are, especially a market like that for bitcoin that is traded 24/7 around the world.

1. Is the lookback window of 60 seconds helpful? What are its limitations? What other features would you want to see in this data set?

Assuming that we are trying to predict price action in the short term, the 60 second lookback could be useful. The issue with the data is that placing trades on such a small interval is very hard. Also, the return on each trade is so small that being off by even a fraction of a second when placing a trade can cause the returns to vanish or even produce a loss. Other features that can help would be additional series based on how the data is changing over time such as moving average and moving volatility over different time horizons.

1. If you stuck with the neural network, what did you change to make it better? Did you change the architecture, did you change the optimizer? The learning rate? The activation function(s)? Why was the model stuck at 0 with an incredibly high root mean squared error?

I stayed with deep learning but used LTSM and GRU as they are much better fitted for time series data. The LTSM (Long Short-Term Memory) can recognize important features, store the time series data in a long-term state, and extract it when it is needed. For these reasons LTSM has been accepted as being very successful at capturing long term patterns in time series data. I also compared it with GRU, which is a simplified version of LTSM that normally performs as well as LTSM, but GRU performed slightly worse in our case.

1. If you used a different model, why'd you choose this model? What about it made it work for this problem? Is this model complex and if so, is the complexity necessary? Is it intuitive enough to explain it to a lay-person? What was your optimizing metric? What were the hyperparameters and why'd you choose them?

I chose LTSM and GRU because, when dealing with time series data, the sequence the data appears is very important. Other types of neural networks take in the same features, but the sequence is lost.

Machine learning models have a general trade off between complexity (and getting a better fit) and the explanatory power of the model, and which model is best will depend on what the project parameters are. In this project, I wanted to get the best fit possible while having the ability to explain the approach taken and why the model I chose is better than other models. However, I would lose the ability to explain which data points are truly driving the model. If I wanted to be able to explain the data points driving the model, other models, such as ARIMA, OLS (with ridge/lasso regularization), and random forest, would be preferred.

For optimization I used Adam because it very efficient at stochastic optimization and has a learning rate per parameter that adjusts as a moving average of the recent gradients. If I had more time, I would optimize the hyperparameters, such as the amount of epochs used, the learning rate, and the dropout rate.

1. Did you include any regularization strategies in your model? If so, why'd you choose the one you did?

I chose the dropout rate, as a regularization strategy, to help reduce overfitting of the data.

1. Did you include visualizations? (everyone loves a good graphic)

I included four graphs in the notebook. For each of LSTM and GRU, I constructed a graph for the predicted vs. realized prices and a graph that combines the training and test data prediction vs. the full data set to get the full picture.

1. How do we know the model is good? How understandable are the diagnostics? How will we know how good the model is predicting in production?

For this model the goal was to minimize the mean squared error. MSE is a very intuitive metric that is very understandable and easily explained to others. The model, after optimizing the hyperparameters, could easily be put into production. Quality of the model’s prediction in production would be determined based on quantum of the returns.

1. If we see data for more than a single day's worth of prices, how do you expect the model to perform? Will it generalize well to new data? Will retraining with this new data be an issue for this model?

If we get data for more than single day’s worth of prices to help train the current model, we can expect improved results. The more data we can get, the more flexible the model will become, which would help reduce variance in the model. However, the more data that we use to train the model, the longer it will take to train the model, so there is a practical trade-off.

1. What question would you ask of the data, or add to this analysis that I haven't thought of?

Is there a way to obtain more data series such as correlated or cointegrated cryptos? What is the plan for implementing a model that is making predictions on a one second interval?