CS349 Spring 2021: Final project

Cryptocurrency price prediction with Long Short-Term Memory (LSTM)

Chanwook Park

0. Reference

For this final project, I referred to the following GitHub and blog posts:

https://github.com/abhinavsagar/cryptocurrency-price-prediction

https://towardsdatascience.com/cryptocurrency-price-prediction-using-deep-learning-70cfca50dd3a

1. Problem statement

In the blog post that I have referred to the LSTM algorithm that gets 5 past ohlcv (open, high, low, close, volume) data and predicts one future price is demonstrated. To understand the procedure, I first run an example with Ethereum's ohlcv data from the past 500 days obtained from Upbit (South Korea's largest cryptocurrency exchange platform) API.

Figure 1 presents the prediction result of test data. It seems our LSTM model generally predicts the price trend well. However, this result is in fact meaningless when we use this for autotrade. Why? It actually follows the curve, not predicts. This is the reason why the prediction curve is slightly left behind the actual curve. This phase lag is also observed at the test result plot in the blog post.

The current LSTM model outputs only the close price of the next day. Therefore, it merely follows the past trend and yields a price that is in line with the past trend. If the market price soars or plunges, we will miss the actual price significantly. Then, how can we solve this issue?

I suggest an architecture of LSTM that outputs multiple successive prices, not a single price. For example, if our LSTM model gets 10 past data and predicts 3 successive future prices accurately, then I will be able to trade for three days in the future with high confidence.

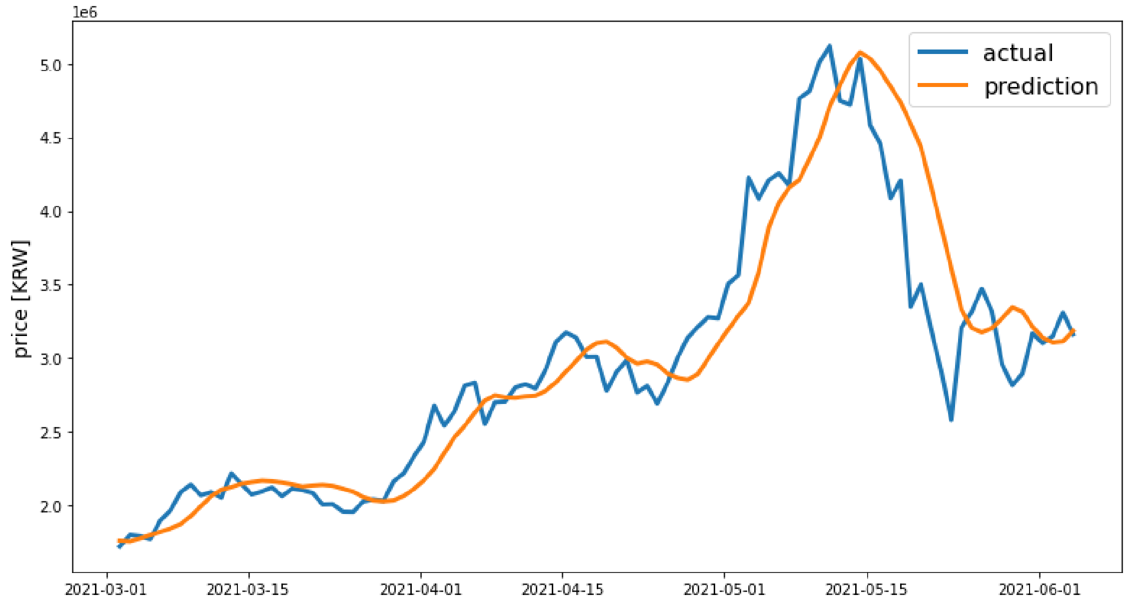

Figure 1 Test results for Ethereum price.

2. Methods

I modified the code in the Github reference to let the LSTM model outputs multiple successive prices. After trying with several initial parameter sets, I set my LSTM model to get 10 past ohlcv data and predict 3 future prices. The code **(LSTM_V3.ipynb)** can be found in the following Github repository:

https://github.com/hachanook/CS349-ChanwookPark

3. Results and discussions

I uploaded two supplementary videos to my git repository: **test.mp4** (test results) and **train.mp4** (train results). Figure *2* presents some snapshots of the test result. The prediction of three successive prices follows the general trend of the curve. However, the pattern of prediction (i.e., three successive prices) does not capture peaks and valleys of the actual data. Instead, it adheres to a continuously descending trend. That is, for most of the predictions, the first-day price is higher than the last one, which makes it meaningless to use this LSTM for autotrade.

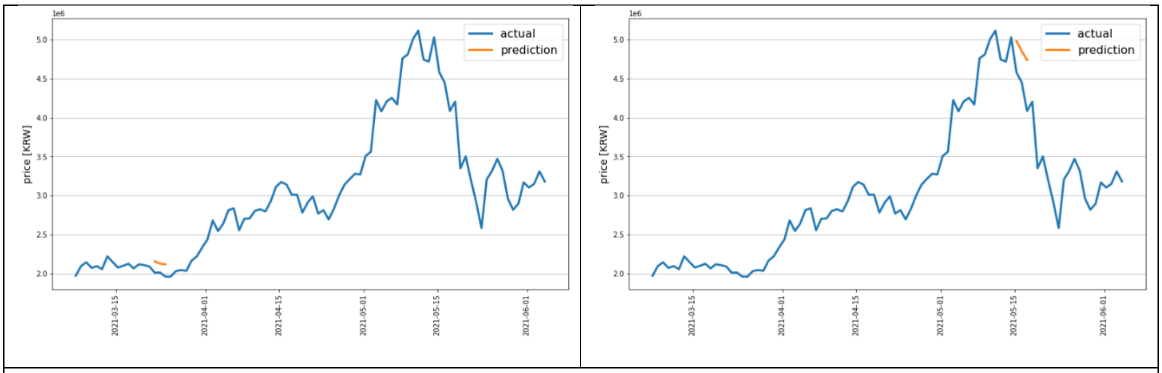

Figure 2 Snapshots of the test result (10 epochs of training).

In my git repository, I also uploaded **train.mp4** that shows the training result. In the video, we can see that the prediction follows the actual curve better than the test result, which implies a possibility of overfitting.

I have used 10 epochs of training. For now, let us train for 5 epochs. Figure 3 shows some snapshots of the test result of the trained model. Unfortunately, adhesion to a stable trend in the prediction data is also observed. In this case, the trend is v-shaped.

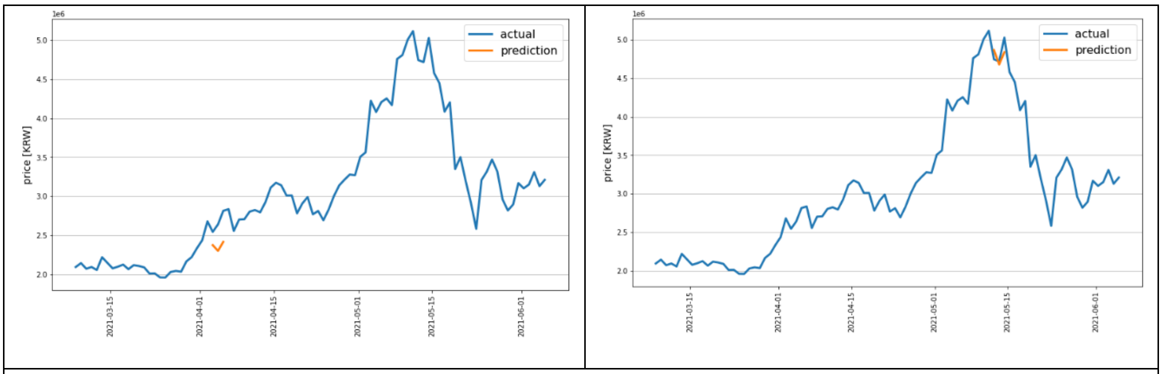

Figure 3 Snapshots of the test result (5 epochs of training).

4. Summary

In this final project, I have worked with the LSTM algorithm to predict cryptocurrency prices. Although the trained model follows the general trend of the actual price curve, it cannot "predict" the future price. There are tons of uncertainties in the cryptocurrency market. Often a "tweet" of an influencer dramatically changes the market trend. Therefore, in order to develop a better model, we need to consider those unmeasurable factors.