BITCOIN PRICE PREDICTION

Oh Young Taek,
Kyung Hee University, Computer Engineering
Suwon, Republic of Korea
Vkdnej91@gmail.com

ABSTRACT

Bitcoin, which is an online virtual currency that is different from the existing currency, is very difficult to predict because of the high price fluctuation. This report introduces the RNN model to predict the price of bitcoin.

1. INTRODUCTION

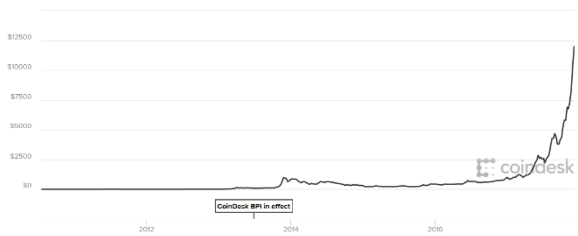
Bitcoin is an online virtual currency created on the basis of block-chain technology.The currency unit of the bit coin is denoted by BTC.

October 2008 Developed by a programmer whose nickname is "Satoshi Nakamoto" , and released the source in January 2009. It is designed to freely transfer money and other financial transactions among individuals in a P2P way around the world without a central bank.

On October 5, 2009, 1-bit coin (BTC) was traded at $ 0.000007563 $ and steadily increased in value, trading at $ 13,000 (16,000,000) at 1BTC as of December 6, 2017(<figure-1>).

As can be seen in Figure 2, the price of bit coin fluctuates greatly. It's not a rational investment, but it's speculative, so people can not analyze the existing data and estimate the approximate price.

Based on the market closing price data up to now, I started the project with the idea that if I apply machine learning on the data, I would be able to predict the approximate price. In addition, the data is time-sequential, so I used Recurrent Neural Network.



<figure-1 : April 2010 - December 6, 2017 Bitcoin Price($)>

<figure-2:
Rapidly collapse from December 4, 2014 and rising from April 20, 2017>

2. MODEL

The dataset [1] used to create the bit coin pricing model is data from April 2013 through August 2017 with <date, start price, best day, lowest price, closing price ($), volume of transaction ($)>

the data is time-sequential. For example, the bit coin price on December 6, 2017 is closely related to the data for December 5, December 4, and December 3.

I used RNN because we determined that RNN is suitable for learning sequence data.

2.1 data preprocessings

1) Rearrange data chronologically

I arrange the data for learning in chronological order.

2) Missing value

Only if the volume is very small, there is no data. So I remove the row.

3) MinMax Scaling to convert values from 0 to 1

$$X = (X - Min) / (Max - Min + 0.000000001)$$

If Max and Min values are the same, we divide by 0, so we divided them by adding very small values to avoid this case.

4) Split train data and predicted data
Because we have to learn based on past data by sequence length, the factors that determine the bit coin closing value of date i are shown in the following table.

date	open	high	low	volume	Close
i-1	value	value	value	value	Value
i-2	value	value	value	value	Value
….	….	….	….	….	….
i - length	value	value	value	value	Value

I also used it 80% of the total data train, and predicted the bitcoin price with the remaining 20% of the data.

2.2 modeling

1) RNN
As I said above, I used RNN because I decided to learn sequential data to predict bit coin price.

2) Drop out wrapper

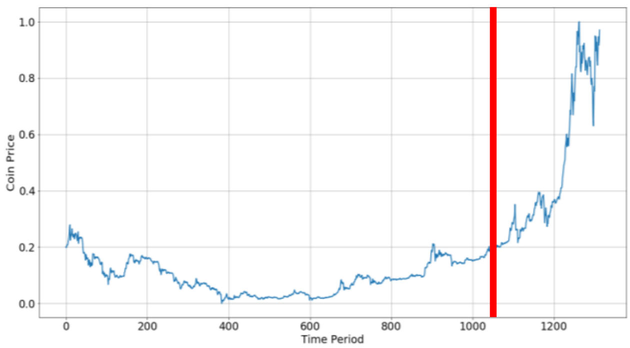
To avoid Over fitting and reduce learning time, I used dropout with 0.7, the rate of drop out

3) Multi-RNN

To make NN more deeper, I used MultiRNN.

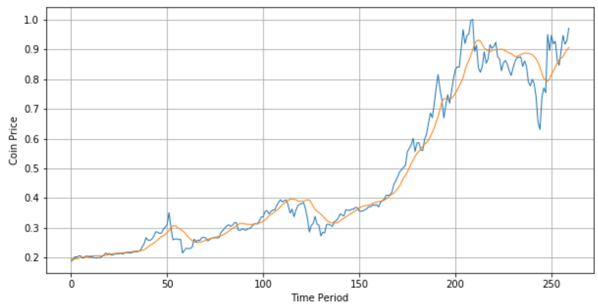
3. RESULTS AND ANALYSIS

The following figure <Figure 3> shows the closing price of the full normalized bit coin price.

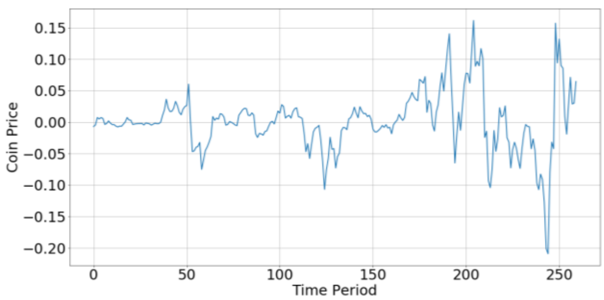


<Figure-3: Bit coin closing price from April 2013 to August 2017, learning the previous data from the red line and predicting with subsequent data>

Details of the model used in the first experiment are as follows. A single RNN cell is used, with a hidden dimension as 100, a sequence length as 15, a learning rate as 0.0001, and an iteration as 500.
The results are shown in <Figure-4> and <Figure-5>.



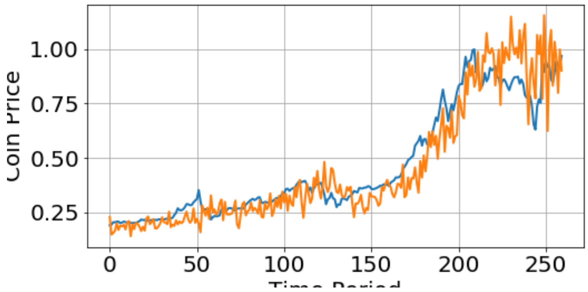
<Figure 4: First experiment, blue line is the actual bit coin price, and orange line is the predicted bit coin price>



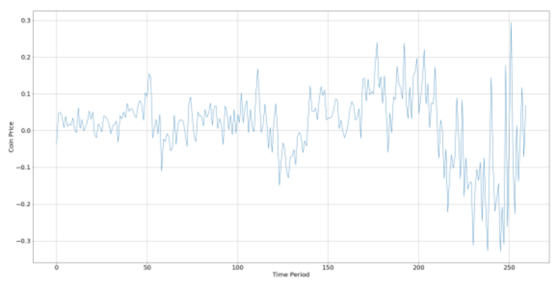
<Figure 5: First experiment, error between actual price minus forecast price>

The results of the first experiment were amazing. As can be seen in Figure 3, the first model seemed to be able to predict an approximate trend even though the price of the bit coin was remarkably different from the learning data, and the price fluctuated irregularly, such as falling. To make a more accurate analysis, we calculated the difference between the actual price and the predicted price, error (Figure 5). The price trend seemed to be somewhat correct, but it can be seen that the error is up to 0.24 in the region where the price rises sharply or falls sharply.

In the second experiment, I used a Multi RNN cell with five layers to make the NN deeper and I gave a dropout of 0.7 and the other conditions were the same. The results of the experiment are shown in <Figure-6> and <Figure-7>. This time, also, it seemed to be able to predict a certain trend (Figure 6), but the difference between the actual price and the predicted price (Figure 7) after time period 150 is similar to the first experiment.In the previous period, we can see that the error is larger than the first experiment.

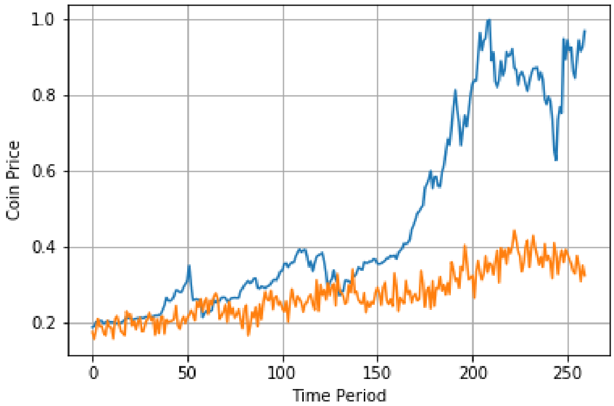


<Figure 6: Second experiment, blue line is the actual bit coin price, and orange line is the predicted bit coin price>

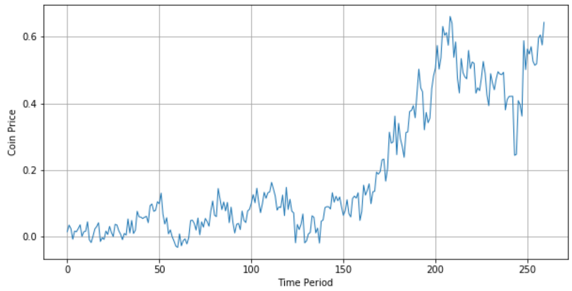


<Figure 7: Second experiment, error between actual price minus forecast price>

In the third experiment, except for using a Multi RNN cell consisting of 10 layers to make it deeper than the second experiment, I proceeded under the same conditions. The results of the experiment are shown in <Figure-8> and <Figure-9>.



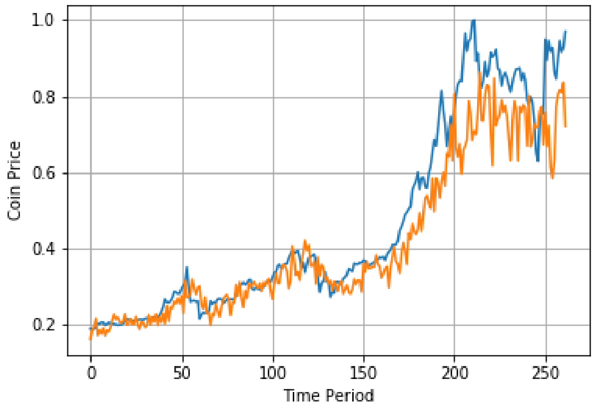
<Figure 8: Third experiment, blue line is the actual bit coin price, and orange line is the predicted bit coin price>



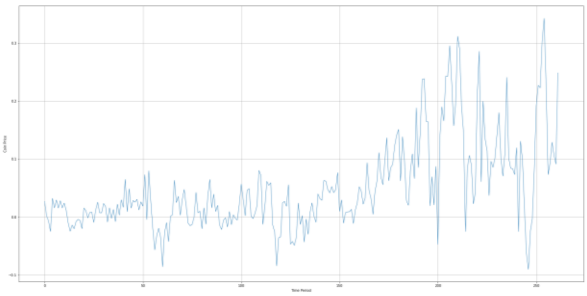
<Figure 9: Third experiment, error between actual price minus forecast price>

This case, it is failed to predict the price trend. A maximum of 0.65 error occurred in the steeply rising section. So far, I have set the sequence length to 15, which I think is not suitable for predicting the price of a bitcoin that changes rapidly.

So I set the sequence length to 7, and I did the fourth experiment. The results of the experiment are shown in <Figure-10> and <Figure-11>.



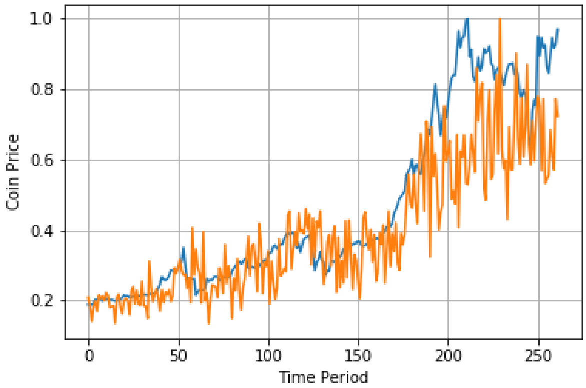
<Figure 10: Fourth experiment, blue line is the actual bit coin price, and orange line is the predicted bit coin price>



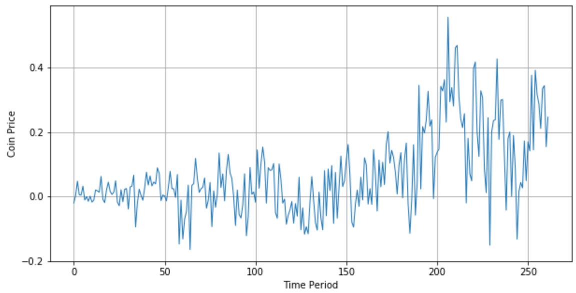
<Figure 11: Fourth experiment, error between actual price minus forecast price>

Compared to the third experiment, we can see clearly better results. In the third experiment, the maximum error is 0.33, compared to the maximum error of 0.65 (Figure 9).

In the fifth experiment, the number of learning iterations was set to 1000 (twice of the fourth experiment) in order to check whether the learning was insufficient because the number of learning iterations was insufficient in the fourth experiment.



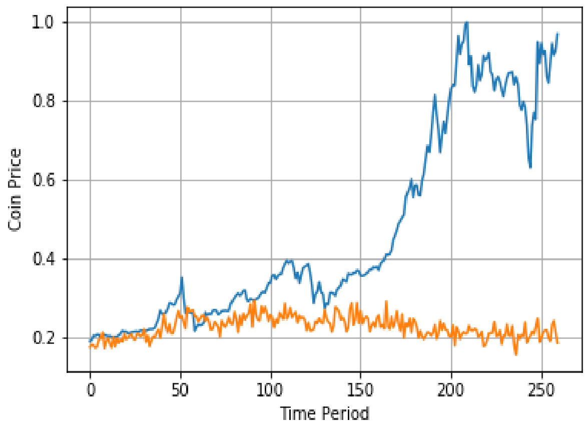
<Figure 12: Fifth experiment, blue line is the actual bit coin price, and orange line is the predicted bit coin price>



<Figure 13: Fifth experiment, error between actual price minus forecast price>

This case, As you can see in Figure 12, it seems to be over-fitting.

In the last experiment, unlike the previous experiments, the hidden dimension was doubled to 200, the learning rate was reduced to 0.00001, 1/10, the learning rate was increased to 10000, and the Multi RNN cell with 20 layers was used . I expected that by reducing the learning rate, making the model more deep and wide, and increasing the number of learning times, better results would be obtained. The results of the experiment are shown in <Figure-14>.



<Figure 14: Lst experiment, blue line is the actual bit coin price, and orange line is the predicted bit coin price>

This case, the learning time lasted more than 12 hours and failed to predict the trend completely, even though loss and RMSE were the lowest in all the experiments. It seems that over fitting to the data of the learning section has not predicted the sudden rise and fall of predicted section.

4. CONCLUSION

Taking all of the above experiments into consideration, the more NN is made deep and wide, the worse the prediction results are. This means that it is not wrong to make the NN deep and wide, but it is impossible to predict the bit coin price using the current dataset. The more precisely you learn the trends of learning data, the more unpredictable the prediction data. <Figure 3> shows that the fluctuation of bit coin price is larger and more frequent in the prediction interval (after the red line) than in the learning interval (before the red line). However, as can be seen in <Figure-4> and <Figure-6>, accurate price prediction is difficult, but some approximate price formation direction can be predicted.

The current bit coin transaction is more speculative than investment. This means that there is big possibilities of irrational and unusual fluctuations of bitcoin price. Starting from the first trading in 2009, the bit coin trading has entered the eighth year since 2017, but unlike the previous one, since the price fluctuation has increased exponentially since March 2017, it is impossible to predict the trend at this point. I conclude that we can use this model to predict the direction of price formation.

5. REFERENCES

[1] dataset from Kaggle, Bitcoin Price Prediction(https://www.kaggle.com/team-ai/bitcoin-price-prediction)

[2] Bayesian regression and Bitcoin - Devavrat Shah Kang Zhang Laboratory for Information and Decision Systems, 2014

[3] Non-linear autoregressive with exogeneous input (NARX) bitcoin price prediction model using PSO-optimized parameters and moving average technical indicators - NI Indera, IM Yassin, A Zabidi, ZI Rizman, 2017

[4] Predicting the price of Bitcoin using Machine Learning - S McNally, 2016

[5] Predict Stock Prices Using RNN,(https://lilianweng.github.io)

[6] Machine learning/ deep learning lecture for everyone, https://hunkim.github.io/ml