CS230

Bitcoin Price Prediction

Ziyuan Lin
ziyuanl@stanford.edu

Sheikh Srijon
srijon@stanford.edu

Jubayer Ibn Hamid
jubayer@stanford.edu

1 Abstract

In this paper, we introduce a deep learning model that predicts the price of Bitcoin. In particular, the model uses an LSTM algorithm trained on data of Bitcoin price since 2nd August, 2012. At the end, the model achieved a trend accuracy of 93.981 percent. The paper also looks into how incorporation of data from other cryptocurrencies such as ETH affects the model's ability to predict Bitcoin prices. Our results demonstrate that using LSTM, deep learning can achieve a notable accuracy in predictions of Bitcoin price.

2 Introduction

Bitcoin is the worlds' most valuable cryptocurrency. The price of Bitcoin has fluctuated greatly in recent years. As a significantly large amount of resources is going into mining and trading bitcoin, this has led to very costly economic uncertainty. Modelling the financial world is interesting because there are a large number of factors that affect each and every change. While it is of course challenging, the benefits of being able to do so are uncountable. We propose a deep learning model for Bitcoin price prediction. The input to our algorithm will be the Bitcoin (and other cryptocurrencies that follow similar trend) price in a number of consecutive days, and the output will be the predicted Bitcoin price on the next day. We will make use of other features such as Exchange Trade Volume, Average Transactions Per Block and transactions per day. We also want to find out whether the information of other cryptocurrencies that have a similar trend as Bitcoin, such as ETH, can be used to improve the model.

CS230: Deep Learning, Spring 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Challenges

There is a large number of factors that affect the price of Bitcoin every single day. On the other hand, the amount of training data we have - if we only limit ourselves to price of bitcoin each day since its inception - is not large. This makes training the algorithm quite difficult. It is also not immediately clear how to get past this constraint on the size of our dataset since data augmentation is quite difficult in this case. We need to find the determinants of the price and properly preprocess them. The result of recent research papers that predicts the price trend is not very good, they built different models but the accuracy has never exceeded 55%, most of the accuracy is between 50%-52%[2]. We can tell the Bitcoin price is fairly arbitrary.

There is a deeper challenge however. We need to research on whether or not an algorithm that can do well on training data from the past and even test data in the present will work well in the future. The factors that determine the price of Bitcoin may also change with time and the weightage of each factor's contribution to the price may alter as well. For our algorithm to be able to predict the price well, we need to be able to find the right set of parameters and determining factors that can help us predict the price for as long as possible. We are taking a closer look at papers such as [6].

4 Dataset

4.1 Data Sourse

The datasets we have are the value of Bitcoin price, including open, high, low, exchange trade volume and percentage change on each day since 2nd August, 2012. In particular, we did not use data from before this date since the price and exchange trade volume before then was particularly low, causing a large amount of noise in the dataset.

We use the sliding window method - we can extract the price (and other features) of m days as the input and the next day's close price will be the expected prediction value. Then we extract the data of recent few months as the test dataset.

4.2 Preprocessing

We dropped the "date" column and normalized the entire dataset. We divided the data across train set, dev set and test set by the ratio 0.70 : 0.15 : 0.15. The data we collected starts from 2010, but we dropped the data before 2nd August, 2012, even though we dropped about 1/6 over all the data we collected, we decided to not use that data because the price and exchange trade volume was too small to give the model any valuable information.

From August, 2012 to May, 2022, we only have less than 4,000 data, the amount of data is fairly small for a deep learning problem. We want to have as much training data as we can but we also don't want to end up with tiny dev and test set, that's why we divided the data with 0.70 : 0.15 : 0.15.

4.3 Labelling

We kept 6 metadata labels for the data - they are close price, open, high, low, exchange rate volume and percentage change.

5 Methods

We used LSTM model for this project which is great at making predictions based on time series data.

The baseline model we use is a one-layer LSTM with 32 hidden units, the window size is 1, and the close price of Bitcoin is the only feature. The baseline loss we get is $5.209 \cdot 10^{-3}$.

After building a baseline model, we started to build a model with two layers. Here, the first layer had 64 hidden units while the second layer had 32 hidden units. When we implemented this model, the prediction result was not very good; the training loss is relatively high. We then resorted to a simpler model. We used a one-layer LSTM with 64 hidden units.

The window size is a very important hyper-parameter to tune. After multiple tests, we settled on the window size 24, as with other window sizes we observed higher test loss and lower trend accuracy. We trained our model with early-stopping and save-best-only (monitoring the dev_loss) with no dropout and the Adam optimizer for faster iterations in our training process.

6 Evaluation

For price prediction, we use Mean squared error (MSE) to evaluate the result. We also consider using price trend accuracy to evaluate our model, i.e. whether the model can predict the correct price trend (price increase or decrease) the next day.

For hyperparameter tuning, we tuned the window size and number of hidden units for the LSTM. At first we fixed the window size as 20, then we trained the model with 16, 32, 64, 128 hidden units, it turned out one-layer LSTM with 64 hidden units fits the dataset the best. Then we fixed the number of hidden units, trained the model with window sizes from size 2 to 30 in increments of 2, the model was sensitive to the window size because that determines how much information to retain in memory, but we didn't find a certain pattern of how the window size affects the model's performance. Experimentally, by doing a loss analysis in dev and test set, We've found the best model is the one with window size of 24, one-layer LSTM with 64 hidden units.

The lowest test loss was $2.873 \cdot 10^{-5}$and highest trend accuracy was 93.98% associated with this model. The trend accuracy is defined as $\frac{pred\_today - actual\_yesterday}{actual\_today - actual\_yesterday}$, we are not trying to improve the trend accuracy during the training, but this is an additional metric that we used to show the performance of the model. The loss and trend accuracy results of all the models are illustrated in the table below:

The following figure shows the loss being minimized with increased number of epochs.

1	Model_windowSize_epochs_units	train loss	dev loss	test loss	trend accuracy
2	model_22_250_64_1	1.09E-07	2.35E-07	[0.00043593]	0.742746615
3	model_24_250_64_1	2.36E-08	4.27E-08	[2.8734876e-05]	0.939805825
4	model_26_250_64_1	1.52E-07	2.13E-07	[0.00010976]	0.923976608
5	model_28_250_64_1	1.95E-07	3.72E-07	[0.00049733]	0.72407045
6	model_30_250_64_1	2.74E-08	4.90E-08	[8.8052075e-05]	0.937131631
7	model_20_300_64_1	2.56E-07	4.42E-07	[8.27702e-05]	0.882466281
8	model_18_300_64_1	2.37E-08	4.53E-08	[9.78481e-05]	0.921305182
9	model_16_300_64_1	2.56E-08	3.15E-08	[6.343035e-05]	0.86998088
10	model_16_350_64_1	2.14E-08	3.58E-08	[9.9823876e-05]	0.910133843
11	model_14_350_64_1	3.17E-08	3.91E-08	[4.6984293e-05]	0.92
12	model_12_350_64_1	4.59E-08	8.16E-08	[0.00071954]	0.732447818
13	model_10_350_64_1	1.95E-07	3.74E-07	[0.00018142]	0.933837429
14	model_8_350_64_1	4.53E-08	6.81E-08	[0.00016847]	0.934086629
15	model_6_350_64_1	5.52E-08	9.63E-08	[0.00019751]	0.936210131
16	model_4_350_64_1	1.82E-07	2.79E-07	[0.00022948]	0.805607477
17	model_2_350_64_1	2.80E-08	3.88E-08	[8.166224e-05]	0.882681564

Figure 1: Loss and trend accuracy of different models

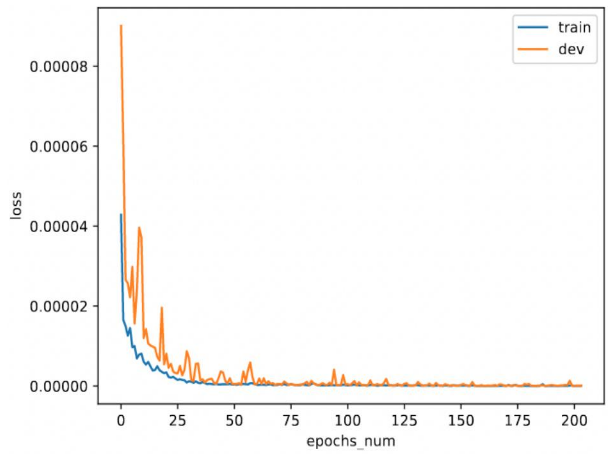

Figure 2: Loss versus number of epochs

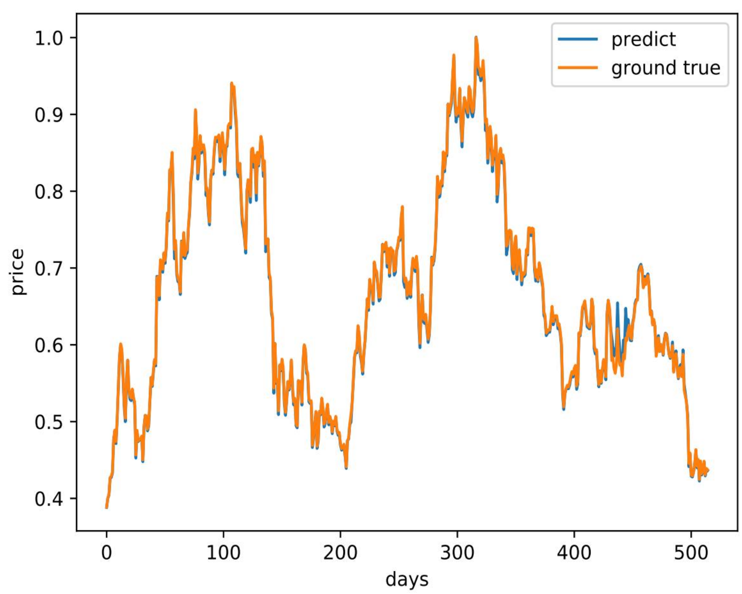

Figure 3: Prediction result of the best model

7 Additional Experiment

To see if we can get better prediction by including prices of another cryptocurrency, we chose to add ETH. Since the ETH data starts from 2016, in order to find how the ETH prices will affect the model's performance, we can only train the model with Bitcoin and ETH data from 2016 onward.

Since we lost about 40% of the data, just training the model with Bitcoin data is not performing as good as before, the test loss is $3.073 \cdot 10^{-3}$. After adding the ETH prices as new features, the test loss is greatly reduced, the final test loss is $1.514 \cdot 10^{-4}$.

For the future investigation, we can use transfer learning, train a model to predict ETH price based on the pre-trained Bitcoin model, we expect to have a great improvement on the ETH price prediction model.

8 Conclusion

In conclusion, we can see that an LSTM model can predict the price of Bitcoin with notable accuracy. Even the dataset is small and the model is simple, we still got amazingly good results, which means the dataset we use doesn't have much complicated information to learn, so we don't need a complicated model, but those information is enough for the model to predict the next day's price based on previous days' data. We would also like to look into other neural network designs. In particular, a combination of CNN and LSTM or Garch and LSTM (as in [7]) seems pretty intriguing.

References

[1] S. McNally, J. Roche and S. Caton, "Predicting the Price of Bitcoin Using Machine Learning," 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018, pp. 339-343, doi: 10.1109/PDP2018.2018.00060.

[2] Ji S, Kim J, Im H. A Comparative Study of Bitcoin Price Prediction Using Deep Learning. Mathematics. 2019; 7(10):898. https://doi.org/10.3390/math7100898

[3] Mingxi Liu, Guowen Li, Jianping Li, Xiaoqian Zhu, Yinhong Yao, Forecasting the price of Bitcoin using deep learning, Finance Research Letters, Volume 40, 2021, 101755, ISSN 1544-6123, https://doi.org/10.1016/j.frl.2020.101755.

[4] Singh Ashish, Kumar Abhinav, Akhtar Zahid, Bitcoin Price Prediction : A Deep Learning Approach, 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Signal Processing and Integrated Networks (SPIN), 2021 8th International Conference on.

[5] Kilimci Hasan, Yildirim Mehmet, Kilimci Zeynep Hilal, The Prediction of Short-Term Bitcoin Dollar Rate (BTC/USDT) using Deep and Hybrid Deep Learning Techniques, 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2021 5th International Symposium on. :633-637 Oct, 2021

[6] Aggarwal Apoorva, Gupta Isha, Garg Novesh, Goel Anurag, Deep Learning Approach to Determine the Impact of Socio Economic Factors on Bitcoin Price Prediction, 2019 Twelfth International Conference on Contemporary Computing (IC3) Contemporary Computing (IC3), 2019 Twelfth International Conference on. :1-5 Aug, 2019

[7] Gao Zidi, He Yiwen, Kuruoglu Ercan Engin, A Hybrid Model Integrating LSTM and Garch for Bitcoin Price Prediction, 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP) Machine Learning for Signal Processing (MLSP), 2021 IEEE 31st International Workshop on. :1-6 Oct, 2021