Machine Learning, Spring 2018
Project2 SSE Composite Index Case Study

Chen Xin	Lin Daquan	Xu Yue
30074765	85610653	61643984

Abstract

In this project, we tried to use two methods to predict the high price and low price. One is LSTM(long short-term memory), a popular model to deal with temporal prediction, in NLP especially. But it seems doesn't work in this project, due to our result is really bad in this model. Then, we utilized a simple model Linear Regression, step is 5, use the prices of Monday in this week to predict the prices of Monday in next week. The performance of linear regression model better than LSTM's. And, our finally predictions is also given by linear regression model.

1 Introduction

In this project, we use the data of the historical SSE Composite Index to forecast the SSE Composite Index for the following week. The SSE Composite Index[1] known as SSE Index is a stock market index of all stocks (A shares and B shares) that are traded at the Shanghai Stock Exchange. Its calculation formula is:

$$Current\ index = Current\ total\ market\ cap\ of\ constituents \times \frac{Base\ Value}{Base\ Period}$$

$$\text{in which}\quad Total\ market\ capitalization = \sum(price \times shares\ issued) \quad (1)$$

2 Model selection

2.1 LSTM

To overcome the error back-flow problem and avoid the long-term dependency problem of RNN, the LSTM(long short-term memory) network[4] training with an appropriate gradient-based learning algorithm. A crucial addition has been to make the weight on this self-loop conditioned on the context, rather than xed[5]. By making the weight of this self-loop gated (controlled by another hidden unit), the time scale of integration can be changed dynamically.In this case, we mean that even for an LSTM with xed parameters, the time scale of integration can change based on the input sequence, because the time constants are output by the model itself[6].

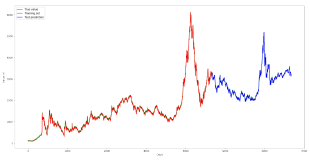

Figure 1: Result of LSTM

2.2 Linear Regression

Since this issue can not get good result in LSTM model, and rethink the features we have. It just a little number of price every day we can utilize. Therefore, a simple model maybe work with this task. At the beginning, we want to use the five days before today(including today) to predict the tomorrow and accept the prediction of tomorrow as prior data to predict the day after tomorrow. And we use average absolute error for five days as the metric.

3 Experiment

3.1 LSTM

Our experimental environment is shown in the Table.1.

Table 1: Experimental environment

OS	Ubuntu 16.04 LTS
Python	2.7.12*
Keras	2.1.5
Tensorflow	1.3.0
Numpy	1.14.2

After loading in the data, discard the two useless parameters of stock code and name. We did not normalize the data. In keras, the LSTM model training batch size is set equal to the predicted step size.[3]

Training for the highest price and lowest price by one step, that means we used the prices of today to predict the prices of tomorrow. Training and validation results are shown in the Fig.1. When carried out prediction, we utilized the prediction prices to predict the prices of next day step by step. It is seems that the results is very bad and low price larger than high price in some days.

3.2 Linear Regression

Like LSTM, data is first preprocessed by dropping those unnecessary features. There are a little number of data equal to 'None', which locates in column

'Turnover'. We just set them as zero, since this number is very small. And normalized the data into scale $[0, 1]$

Since "5 Day Moving Average" can be used to buy stocks[2], but in there, we supposed each day has different weight, the price of today may has largest weight, if we forecast the price of tomorrow, intuitively.

After completing the training, we checked our model in validation date range from May. 28 to Jun. 1. Our predictions are shown in the following Table.2.

Table 2: Result of Linear Regression

	Highest price estimate	Highest price	Lowest price estimate	Lowest price
May. 28	3228.2090	3149.6646	3181.7468	3115.9585
May. 29	3213.3618	3143.2075	3171.7676	3112.153
May. 30	3169.1458	3085.397	3128.4106	3041.0002
May. 31	3157.9163	3098.0764	3120.7886	3054.2686
Jun. 1	3149.2256	3102.088	3107.4355	3059.7856

average absolute error for five days is 30.92, 29.81 respectively.

4 Result

Under the LR model and the LSTM model, we expect the index for the next five days to be as shown in the table.3.

Table 3: Prediction of the Index from June 4 to June 8

Type	June 4	June 5	June 6	June 7	June 8
Low Price	3096.836	3087.8623	2996.0542	3067.379	3032.6873
High Price	3140.6753	3131.1768	3035.664	3110.1226	3078.4883

References

[1] https://en.wikipedia.org/wiki/SSE_Composite_Index

[2] https://www.investopedia.com/articles/active-trading/052014/how-use-moving-average-buy-stocks.asp

[3] https://stackoverflow.com/questions/43702481/why-does-keras-lstm-batch-size-used-for-prediction-have-to-be-the-same-as-fittin

[4] Hochreiter, Sepp and Schmidhuber, Jürgen, Long short-term memory, *Neural computation*, 9(8):1735-1780, 1997

[5] Gers, F. A., Schmidhuber, J., and Cummins, F, Learning to forget: Continualprediction with LSTM. Neural computation, 12(10): 24512471.2000

[6] Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016