CSI 300 Prediction Using LSTM Model

Jiaming Mai

Finance Institute, Guangdong University of Foreign Studies, Guangzhou 510000, China

20171704749@gdufs.edu.cn

Abstract. LSTM is used in the article to forecast CSI 300, and consider if adding volume (the number of shares transacted every day) and p_change (amount of increase and amount of decrease) on the basis test (the common variables include open, close, high and low price) will have a better result; The result is also compared with predictions using SVR and the GBDT model, and the MSE of the LSTM and SVR test are lower than GBDT model.

Keywords: Stock Price Forecast; LSTM; MSE; CSI 300.

1. Introduction

Stock price forecasting has always been one of the popular research directions in the field of financial statistics. But the stock market in China is extremely volatile and instability due to some policies or other reasons, making it difficult for shareholders to invest in stocks for profit. Many scholars have made more in-depth research on the influencing factors of stock prices and stock price forecasting techniques. Financial market forecasting methods are totally divided into forecasting methods based on time series and those based on machine learning. As for methods based on time series, people usually use liner models such as AR, ARMA, ARIMA, and non-liner models include ARCH and GARCH. For example, in 2016, Mohammad used the ARMA model to forecast the price of S&P 500[1]; In 2017, empirical stock returns' rolling windows are used to test the GARCH model in Herwartz H's paper [2].

However, the high noise and uncertainty of stock data make it difficult for traditional time series forecasting models to forecast accurately. Machine learning have gradually replaced time series models in financial forecasting. At present, the prediction methods of machine learning mainly include support vector regression machine(SVR) and Gradient Boosting Decision Tree(GBDT). Ding Z predict the stock price trend with SVR and Support Vector Machine Regression has shown a good result throughout the test [3]. the GBDT algorithm which is an integrated learning algorithm using the idea of Boosting was proposed and improved by J. Friedman [4].

However, if the dimension of the data is too high, the results of SVR test will be affect. At the same situation when using GBDT, it will increase the complexity of the computing process. Therefore, this article wants to use existing efficient models to predict stock prices, so that normal investors can predict stocks themselves. Stock prices can be predicted theoretically, but there are many factors that affect stock prices, and so far their impact on stocks has not been clearly defined. This is because stock forecasts are highly non-linear, which requires the forecasting model to be able to deal with non-linear problems. Stocks are change irregular so it is suitable to use cyclic neural networks to predict stocks.

Although the cyclic neural network (RNN) allows information to be more persistent, the general RNN model has a weak ability to describe time series data with long memory. If time series is too long, training with RNN model will become very difficult and will not work so well. Therefore, LSTM model comes out to be a more effective and better method. Hochreiter and Schmidhuber first proposed this model in 1997[5].Fischer T finds that LSTM networks model to outperform memory-free classification methods[6].Honchar O and Persio L D study S&P 500 data in 3 different methods: MLP, CNN and LSTM and conclude that a novel approach based on combination of wavelets and CNN which outperforms basic neural networks approaches[7].The LSTM model in deep learning can efficiently describe the long memory of time series. The long short-term memory model (LSTM) is

a special RNN model to solve the problems in RNN. Special gate mechanism that RNN model does not have is introduced to this model to solve the long memory problem.

2. Methodology

Following will describe the working mechanism. As shown in Figure 1:

Figure 1. LSTM mechanism

There are 3 types of special gate of LSTM model: forget gate, input gate and output gate. These valves can be opened or closed, and are able to judge whether the output result of the memory state of the model network in this layer reaches the certain value and then add it to the calculation of the current layer.

2.1 Forget Gate Layer

Figure 2. Forget gate

This gate is to distinguish and decide what information should be discarded from the cell state. Read the output (h_{t-1}) of the previous layer and the input (x_t) of the current layer. The output result is a value from 0 to 1. as for the number in the cell C_{t-1} , 0 means abandon completely and 1 means completely reserved. The formula process of this layer is as follows:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \quad (1)$$

2.2 Input Gate Layer

Figure 3. Input gate

This gate determine what information will be updated. There are two parts: one is the “sigmoid” function layer to determine what information should be input, and the other part is “tanh” function layer, which create a new candidate value vector, which will be added to the state. The formula process of this layer is as follows:

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \quad (2)$$

$$G_t = \tanh(W_c \times [h_{t-1}, x_t] + b_c) \quad (3)$$

2.3 Update Gate Layer

Figure 4. Update layer

It is a layer which updates the old and new cell states from C_{t-1} to C_t . The formula process of this layer is as follows:

$$C_t = f_t \times C_{t-1} + i_t \times C'_t \quad (4)$$

2.4 Output Gate Layer

Figure 5. Output gate

The last gate can determine what results should be output. And This output is based on the cell state: First a “sigmoid” layer is running to determine which part of the cell state will be output; Then through “tanh”, the cell state is processed and multiplied with the output of the “sigmoid” gate layer to decide the output part. The calculation formula of this layer is as follows:

$$O_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = O_t \times \tanh(C_t) \quad (6)$$

3. Experiment

The training of the experiment is conducted by the following procedures.

3.1 Data Collection

We acquired the historical CSI 300 data by tushare package. It has 304 daily records of 1824 data from 2015/01/01 to 2020/01/01, involved six variables: open price, high price, close price, low price, trade volume. Each variable contains one column of data. Use this data for training and testing, and

predicting the second day based on the previous day's information. The programming environment is Python3.7. Experiments are using the Keras package.

3.2 Data Normalization

Since the stock price and trading volume are input as feature values at the same time, there are generally hundreds of millions transaction volume, and the price is much lower (mostly tens of hundreds). It cannot be shown that the value of the trading volume is too large. The impact of price is large, so it is necessary to normalize the feature sequence.

3.3 Training Detail

In this experiment, two hidden layers are set. "batch_size" (the number of samples for one training) is set to 100, epoch (the number of training times) is set to 100, and the time step is 1. MSE is used as the loss function. The hidden layer activation function chooses to "relu"; And using the "Adam" to optimize. Choosing "Keras" as deep learning platform. The codes are as following:

```
d = 0.0001
model = Sequential()
model.add(LSTM(64, input_shape=(window, feanum), return_sequences=True))
model.add(Dropout(d))
model.add(LSTM(16, input_shape=(window, feanum), return_sequences=False))
model.add(Dropout(d))
model.add(Dense(4, init='uniform', activation='relu'))
model.add(Dense(1, init='uniform', activation='relu'))
model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train, nb_epoch=100, batch_size=256)
```

4. Results and Discussion

4.1 The Affect When Adding Volume and P_Change When Using LSTM

Put the open price, close price, high price, low price into the training. The results are shown in Figure 6. the first figure shows the comparison of the prediction of the training set and the fact result of the set. It can be seen that the trend is basically captured and the prediction is quite good. The second figure demonstrates the prediction the test set comparing with the real results. but it can be seen that there will be deviations.

Figure 6. Result of four features test

Adding volume and p_change (amount of increase and amount of decrease) to the training. The results are shown in Figure 7. The first figure shows the comparison of the prediction results of the training set and the real results. It can be seen that the trend is basically captured and the prediction is also very good. the second t figure shows the comparison of the predictions data of the test set and the real data of the test set. It can also be seen that there will be deviations, but the rising and falling trends have been fully captured.

Figure 7. Result of six features test

The MSE summary is obtained as follow. it can be found from Table 1 that the prediction results of the six features training have the lower accuracy rate, that is, whether it is the training set or the test set, MSE is significantly higher than four features' cases. The lower the MSE, the more accurate prediction results are. Therefore, it can be concluded that adding volume and p_change will make the prediction effect more inaccurate.

Table 1. MSE with different variables

VARIABLE	TRAIN MSE	TEST MSE
the origin four variables	0.0033015284	0.0039180020
six variables when adding volume and p_change	0.0057942383	0.0042421344

4.2 Comparison of the Results Via Using LSTM with SVR and GDBT

Based on the LSTM model, the experiments will be compared with SVR and GDBT, and both use six attributes as features. Following are Figure 8 and Figure 9 about the predictions when using SVR and GDBT.

(a)

Figure 8. Result of SVR

Figure 9. Result of GBDT

It can be found from Table 2 that the prediction results of the six features training have the lower accuracy rate, that is, whether it is the training set or the test set, MSE is significantly higher than four features' cases. The lower the MSE, the more accurate prediction results are. Therefore, it can be concluded that adding volume and p_change will make the prediction effect more inaccurate.

Table 2. MSE with different model

MODEL	TRAIN MSE	TEST MSE
LSTM	0.0057942383	0.0042421344
SVR	0.0014971852	0.0034456030
GDBT	1.5579361624	4.7288113188

We can easily know from Table 2 that the test set of GDBT has the highest MSE, and the MSE of SVR and LSTM have achieved better prediction results. It shows the effectiveness of both LSTM and SVR.

5. Conclusion

The prediction of stock price or index has always been popular in recent years. In this article, we adopt the LSTM model and collect the data of CSI 300 from January 1, 2015 to January 1, 2020 for a total of 304 trading days for training and testing. The results show that adding volume and p_change will not make the prediction result more accurate. what is more, the result is better than the GDBT models and is as accurate as SVR.

Based on the different advantages of different models, maybe we should choose the model in different situations, trying to increase the accuracy of forecasting. Even consider adding some news relating to stocks market or other effective features to train the model again with more accurate results.

References

- [1] Mohammad, Mahdi, Rounaghi, et al. Investigation of market efficiency and Financial Stability between S&P 500 and London Stock Exchange: Monthly and yearly Forecasting of Time Series Stock Returns using ARMA model [J]. *Physical A: Statistical Mechanics and its Applications*, 2016. DOI: <https://doi.org/10.1016/j.physa.2016.03.006>.
- [2] Herwartz H. Stock return prediction under GARCH --empirical assessment[J]. *International Journal of Forecasting*, 2017, 33(3): 569-580.DOI: <https://doi.org/10.1016/j.ijforecast.2017.01.002>.
- [3] Ding Z. Application of Support Vector Machine Regression in Stock Price Forecasting[M]. Springer Berlin Heidelberg, 2012. DOI: https://doi.org/10.1007/978-3-642-27966-9_49.
- [4] Ridgeway G. Discussion of Additive logistic regression: a statistical view of boosting (J. Friedman, T. Hastie, R. Tibshirani, auths.). 2000.
- [5] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. *Neural Computation*, 1997, 9(8):1735-1780. DOI: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [6] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions [J]. *European Journal of Operational Research*, 2017, 270(2). DOI: <https://doi.org/10.1016/j.ejor.2017.11.054>.
- [7] Honchar O, Persio L D. Artificial neural networks approach to the forecast of stock market price movements. 2016.