Stock Prediction using LSTM

Xiangyuan Ren
UCSD
xir010@eng.ucsd.edu

***Abstract*—In this report, I will talk about my experiment on the stock dataset, and understanding of the paper Stock Price Prediction via Discovering Multi-Frequency Trading Patterns[2], which is call SFM. In the implementation section, I will describe how I used LSTM, which is less important.**

I. SFM

A. Introductio

In the paper, the Dataset contains 50 different stock prices among 10 sectors from 2007 to 2016.

The main two ideas for the author is: a) combine multiple stocks to train one model, so different stocks can affect each other. b) add frequency term, so the model can predict long term or short term by including corresponding frequency.

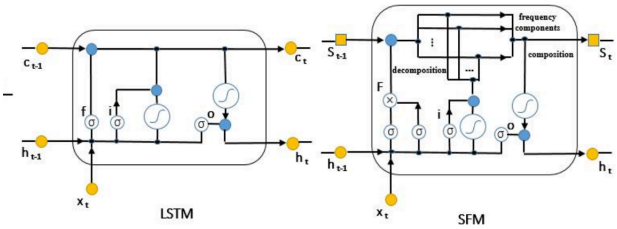

Fig. 1. LSTM vs SFM

As is shown in Fig.1. the SFM includes a frequency term that used DFT to split the state into multiple frequency and then uses IFT to inverse.

B. Failure

Though it is "successful" because of its acceptance of KDD 2017, the model does not work.

As is shown in Fig.2. the prediction is very close the real price. However, it can also be interpreted as a "delayed version" of real price.

For stock prediction, the model should tells whether the stock will rise or fall. the SFM result is always one day after the reality, which means that its prediction is one day after the reality, which undoubtedly means nothing.

Fig. 2. SFM Prediction

The main reason for its failure is that the metric is inappropriate. The author used MSE to measure the performance of a model. But what we need something different. Alarm is what we need instead of a approximation.

Now, my suggestion is to use reinforcement learning, which update according to the changes in the reality. The truth is that, the stock price is more than a feedback from the past, it is a game theory that those who act faster win more.

II. LSTM

A. Problem

I used LSTM for this dataset, since I think I should first try SFM with all the frequency included. The two main problems are: a) I haven't learned deep learning before. b) I haven't used Keras before.

Thus, I first went to get a certificate from Coursea to learn deep learning and then implemented a simple version of LSTM on the International Airline Passengers prediction problem[1].

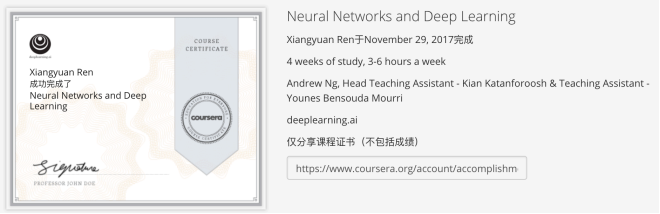

Fig. 3. Coursea Certificate

B. Result

After implementing LSTM on single stock prediction and multiple stock prediction, I found that single stock prediction will always yield a "delayed version" of reality. Although this is not less accurate than SFM in term of MSE, they are almost the same.

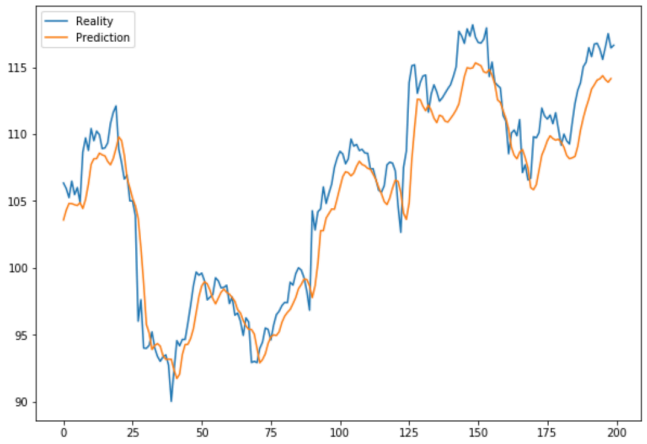

Fig. 4. Single Stock result

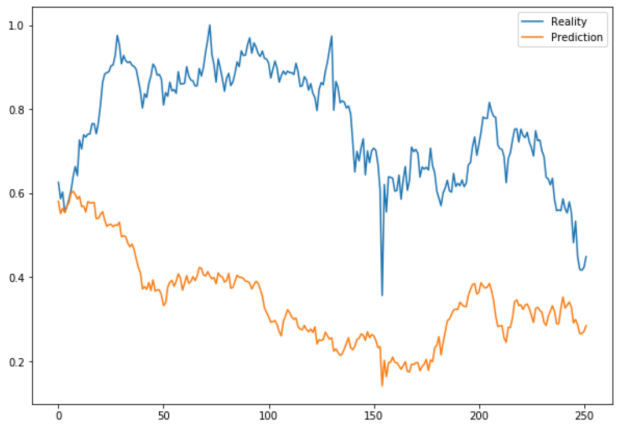

Fig. 5. Multiple Stock result

However, the multiple stock prediction, which can sometimes yield a look-ahead version of reality, though it is much worse in term of MSE. I think it is mainly because of the mutual influences of stocks. For example, the stock A affected by storm two days before stock B, and such influence from A gives a look-ahead alarm for the prediction on stock B.

C. Airline Implementation & Parameters

The Airline Implementation is something different, the author manually shift the data so every input data consists of multiple timestep, instead of using the time step parameters, I think it is due to the Keras bad document. Here I want to say that for LSTM, the input should be the representation of the single data, for example, a 10 dimensional vector that embedding the word vector. The attribute of the time series should be included by using LSTM's time step parameters. This really cause me much time to understand.

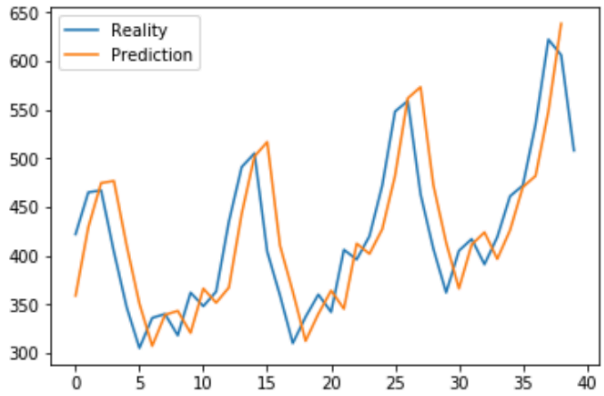

Fig. 6. Airline result

III. Conclusion

Method	Describe
LSTM without timestep	simple RNN
Single Stock with timestep	accurate but no alarm
Multiple Stock with timestep	not accurate but has alarm
SFM	accurate but no alarm

References

[1] International airline passengers prediction problem. http://blog.csdn.net/aliceyangxi1987/article/details/73420583.

[2] GJ Qi L Zhang, C Aggarwal. Stock price prediction via discovering multi-frequency trading patterns. http://www.kdd.org/kdd2017/accepted-papers, 2017.