Complementary Work: LSTM and Rolling Windows Linear Regression

We used LSTM and Rolling Windows Linear Regression as two complementary methods to support the research of NN and HMM.

However, the more our group digging into these two models, the more we believe that LSTM and Regression models can have a better theoretical result.

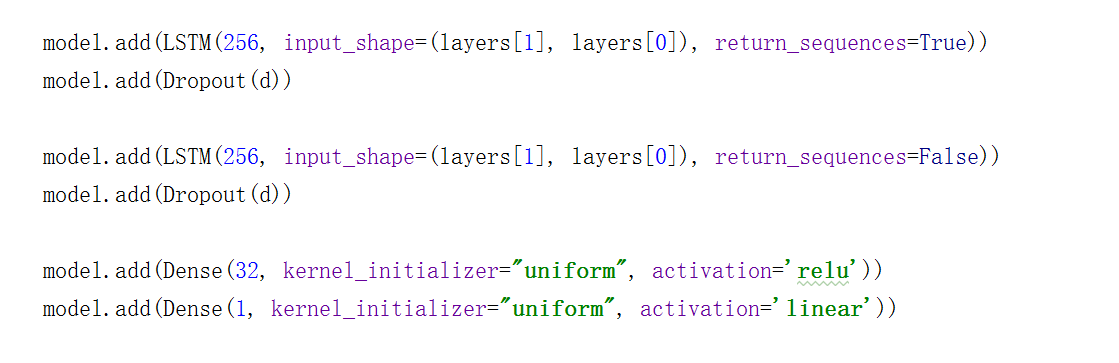
NN and especially CNN is good at feature extraction. This means that if we give CNN a picture, CNN can extract the contour of the picture. However, we need to use RNN and LSTM for prediction because they are for sequence prediction and we all know stock is strongly related with sequence.

The reason that we choose LSTM but RNN is that RNN has simple structure. RNN has one tanh layer and two interactive layers, but LSTM has four tanh layers and four interactive layers. By using the memory cell, we can make sure that we don’t have vanishing gradient problem, which means we can have a good prediction.

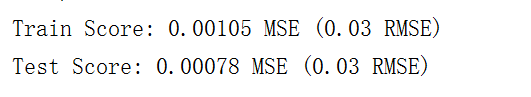
Because LSTM has really a similar structure as CNN, we are not going to mention everything about the code. However, the procedures are below:

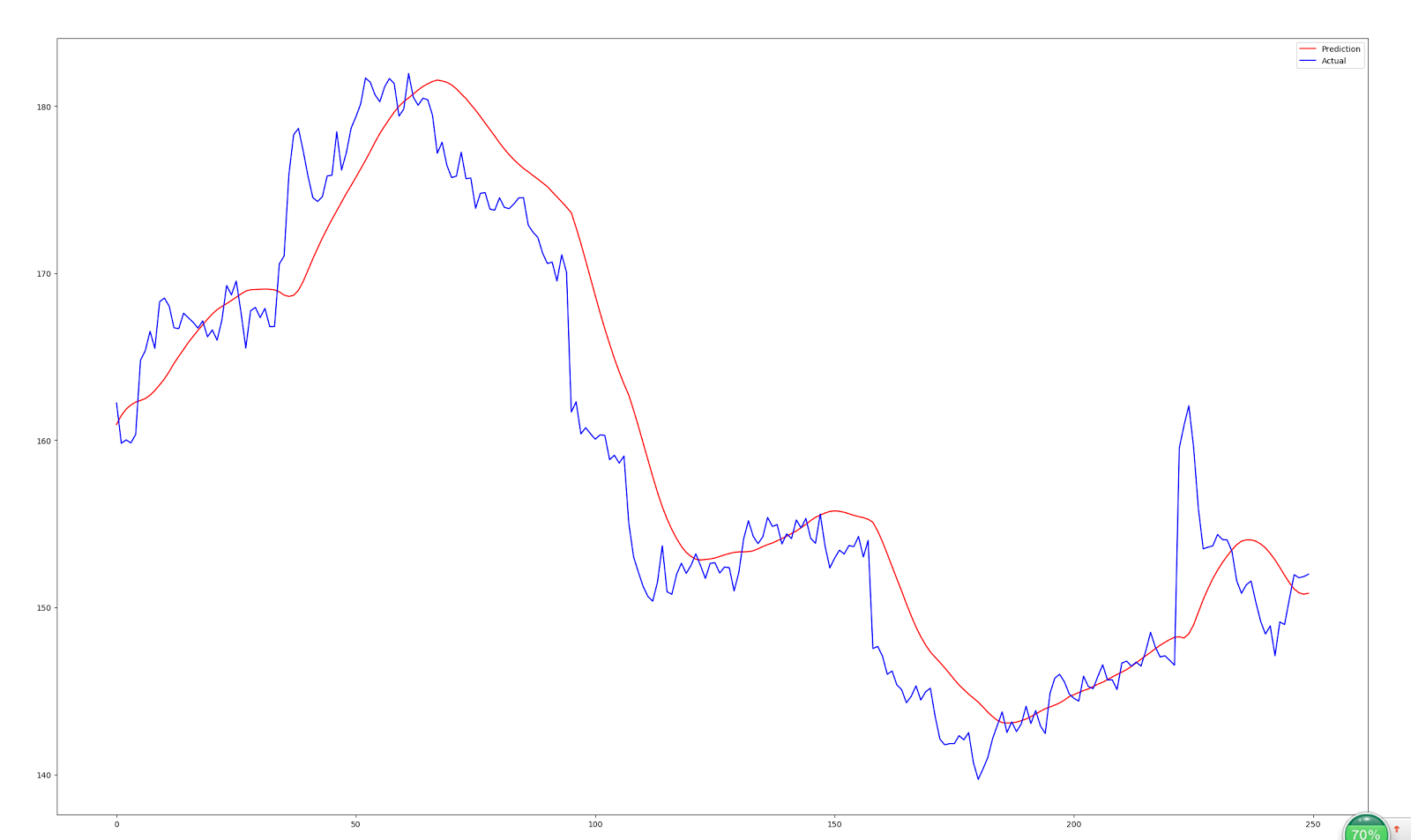
We used open, close max and min price for training our model.

We have windows=22, which means we can satisfy the professor’s requirement. We use 22 days for training and 22 days for testing, and another 22 days for training and another 22 days for testing and so on so forth.

We load the data. Normalized the data to (-1, 1). Build the model with two LSTM layers and one fully-activation layer.  Last, denormalized the data.

The score and the result are below:





From here, we can see that the predicted value (red) and actual value (blue) are similar. The predicted value can reflect the actual value’s trend. However, there are some kind of delay in time, which is a minor in this prediction. However, we use only epoch=30, if the epoch changed to 100, the problem can be compromised. I am sorry I only have a cpu but gpu.

Rolling Windows Linear Regression can be built by an Office Excel. Therefore, the code is actually not very hard. Like the definition of the rolling windows, we use windows=22 days for training, and use another windows=22 days for training, and so on so forth.

The procedures are below:

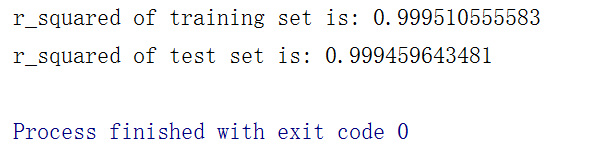
We used open, close, max and min price for training.

We used open, max and min prices for matrix X, which is a matrix of feature and bias.

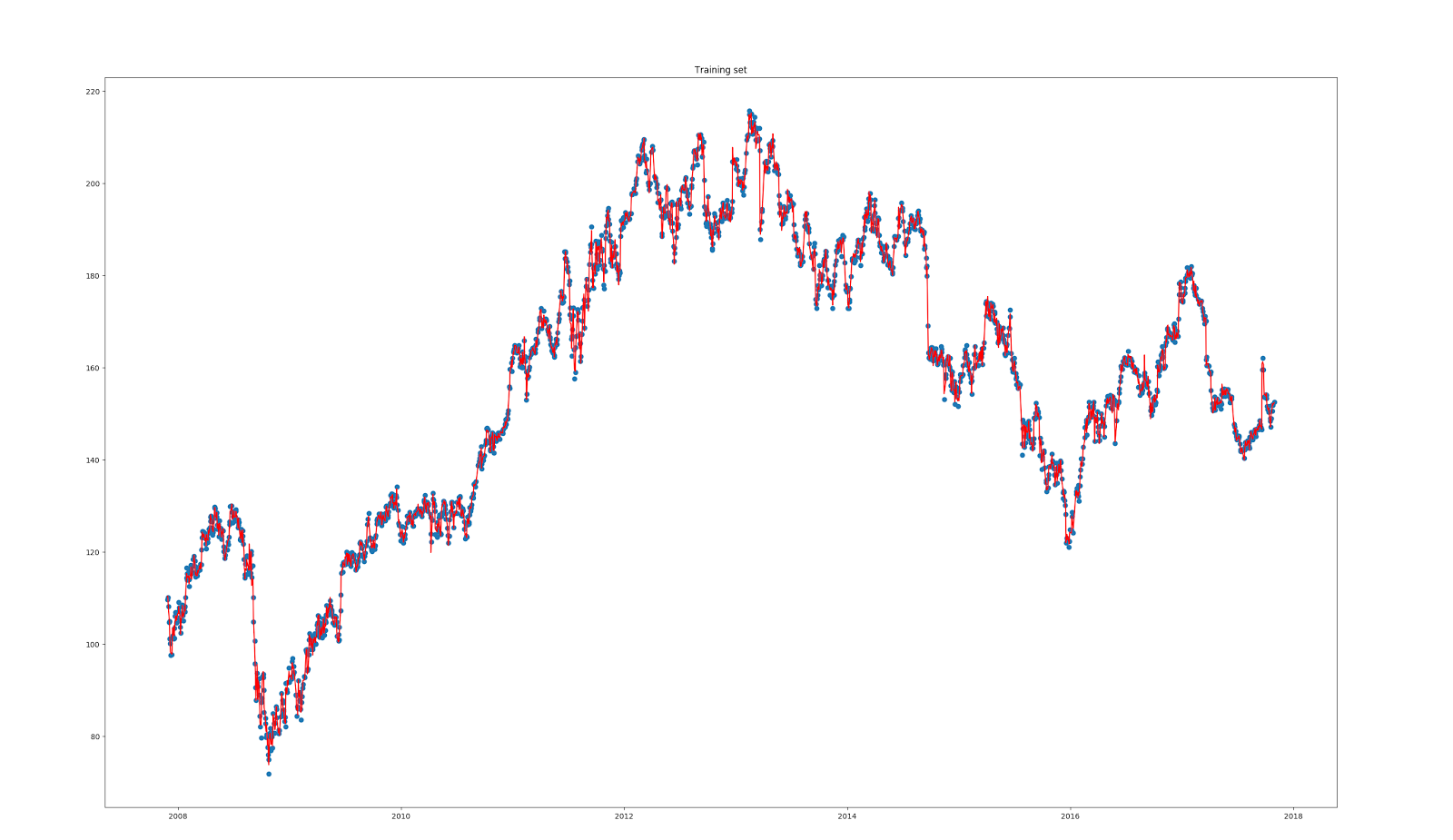
We used close prices for matrix Y, actual output value.

After using X and Y for building up training and testing data, we calculated the weights from training set and used that for testing set.

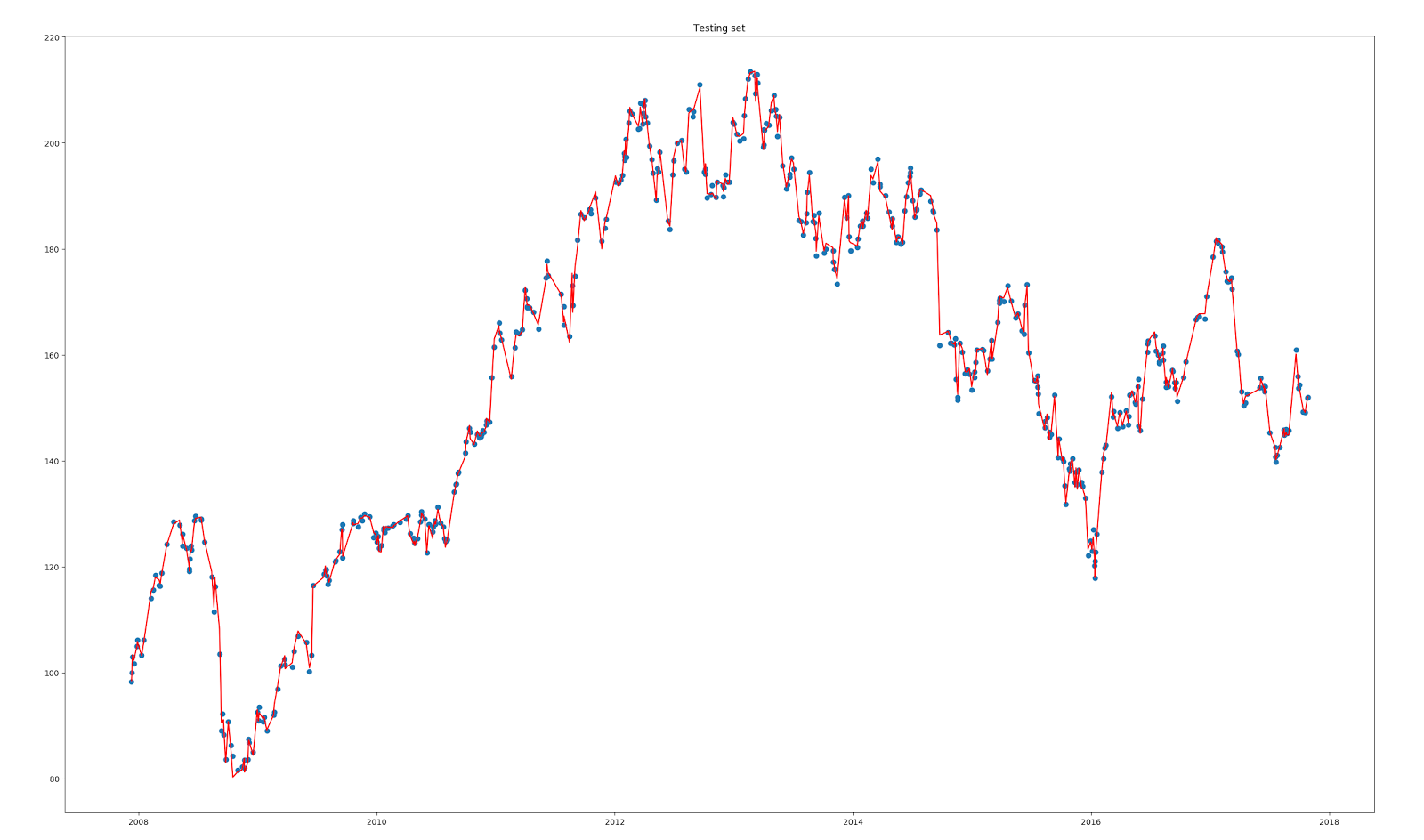
The result is shown below:



R\_squared is nearly 1, so you can see that the actual value and the predicted value are closely related.



In training set, the randomly chosen actual values are crossed by predicted value. Nearly every points is crossed.



In addition, in testing set, the randomly chosen actual values are crossed by predicted value.

Overall, theoretically, LSTM can predict the trend of the actual stock price in a descent way. So is the linear regression. However, we don’t know for sure why Linear Regression can converged in so short a time.