Deep Learning in Long-Term Price Prediction

Keywords: LSTM, RNN, Three-class predication, Loss function

By Xiangwu Li.

July 25th, 2018

I. Overview

1. Introduction

Predicting stock price trends has always been an attractive topic to both investors and researchers. A variety of analysis methods have been developed, such as fundamental analysis, technical analysis, quantitative analysis, and machine learning is no exception. Among those attempts to predict stock price with machine learning, most researches focus on price prediction or two-class price trends prediction (which is, either rise or fall). But two-class classification may not give an effective signal for trading because some very slight fluctuations around the former price are not worth considering for selling or buying. In this project, I focused on predicting the next rolling period's returns / price trends respectively with recurrent neural network (RNN) and Long Short-Term Memory (LSTM), especially **three-class prediction for long-term price trends**, finally achieving an out-of-sample accuracy of **73%**. (code and data of this project can be found at https://github.com/Luffy-wu/LSTM-Stock-Predition .)

2. Model Description

a. RNN

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed cycle. RNNs can use their internal memory to process arbitrary sequences.

The basic architecture can be presented as follows; at each time step, the input is propagated and then a learning rule is applied, integrating the new information and the previous stored in the hidden units. Hence, the network can maintain a sort of states, capable of performing such tasks as sequence-prediction that are beyond the power of a standard multilayer perceptron.

$$h_t = \sigma_h(W_h * x_t + U_h * h_{t-1} + b_h)$$

$$y_t = \sigma_y(W_y * h_t + b_y)$$

Variables and functions:

x_t: $input\ vector$

h_t: $hidden\ layer\ vector$

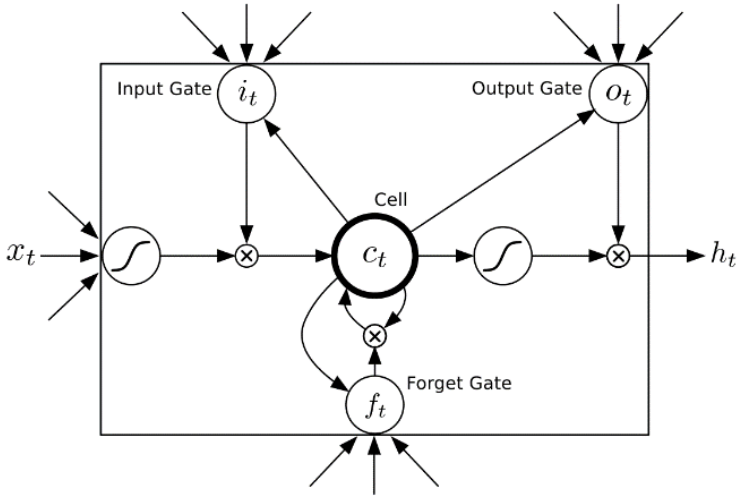
y_t: $output\ vector$

$W, U\ and\ b$: $parameter\ matrix\ and\ vector$

$\sigma_h\ and\ \sigma_y$: $activation\ function$

b. LSTM

A Long Short Term Memory network (LSTM) is a special kind of RNN, capable of learning long-term dependencies. Compared with RNN, LSTM introduces three gates and one memory cell to control the flow of information into or out of its memory. The architecture can be presented as follows:



An "input gate" controls the extent to which a new value flows into the memory. A "forget gate" controls the extent to which a value remains in memory. And, an "output gate" controls the extent to which the value in memory is used to compute the output activation of the block. These allow LSTM to store, forget, and read information from the long-term state of the underlying dynamics.

$$i_t = \sigma_i(W_i * x_t + U_i * h_{t-1} + b_i)$$
$$f_t = \sigma_f(W_f * x_t + U_f * h_{t-1} + b_f)$$
$$o_t = \sigma_o(W_o * x_t + U_o * h_{t-1} + b_o)$$
$$c_t = f_t * c_{t-1} + i_t * \sigma_c(W_c * x_t + U_c * h_{t-1} + c)$$
$$h_t = o_t * \sigma_h(c_t)$$

Variables and functions:

x_t: $input\ vector$

h_t: $output\ vector$

c_t: $cell\ state\ vector$

$W, U\ and\ b$: $parameter\ matrix\ and\ vector$

f_t: $forget\ gate\ vector$

i_t: $input\ gate\ vector$

o_t: $output\ gate\ vector$

$\sigma_i, \sigma_f, \sigma_o, \sigma_c and\ \sigma_h$: $activation\ function$

II. Data Exploration

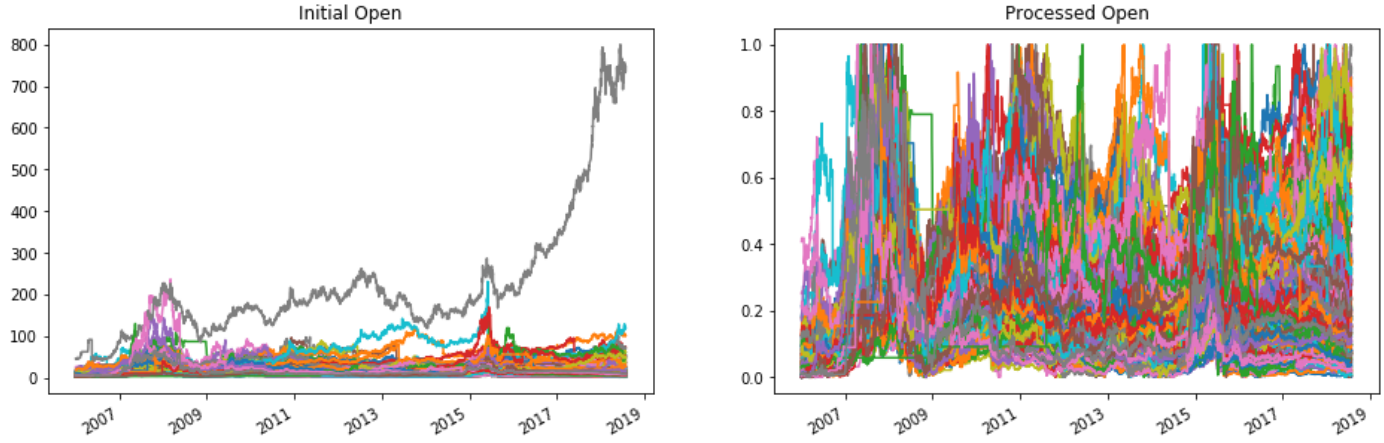
1. Data Collection

The original datasets included the open, high, low, close, volume, amount, percent change, turnover of all the constituent stocks of CSI300 index from Jan.1st 2006 to Jul.31st 2018, sourced from Wind.

2. Data Processing

- Removed stocks that had been suspended for more than 90 days
- Increased data by transformation, final dataset dimension was 35500*30*8
- Normalized trading data
- Divided it into training & test set and set labels, shuffled it

Initial and Processed Data (e.g. the Open)



III. Experiment and Result Analysis

1. Architecture

In my experiment, I used three-layer RNN and LSTM, each layer having size of 32 hidden units. The first two layers were sequence to sequence layer and the third layer was sequence to output. I chose Mean Squared Error (MSE) with L^2-regularization on the weights for the cost function:

$$L(\theta) = \frac{1}{m}\sum_{t=1}^{m}(y_t - \hat{y}_t)^2 + \lambda \sum \|W_i\|^2$$

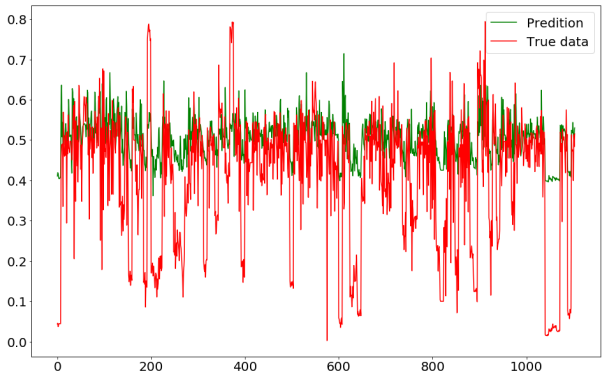
I got 35500 sequences from Jan.1st 2006 to Jul.31st 2018. From these I used 34000 samples for training and 1500 ones for validation, with a various set of parameters and different epochs to measure the MSE of training and testing dataset.

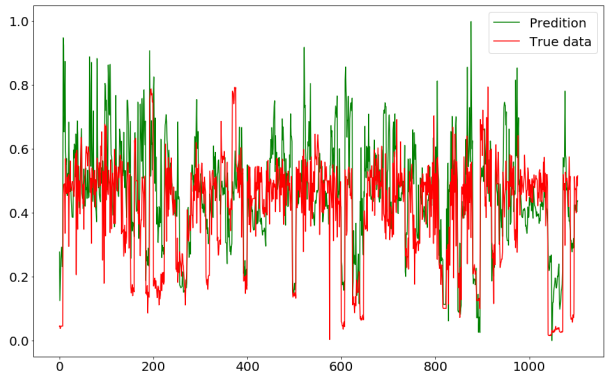
2. Predicting Returns

At first, I used market data of last 30 days to predict average returns in the next period. The left in following figures was to use one of market data as input and the right was to use all 8 types as input. It showed

that it gave a more accurate prediction when using more input information. To improve it, then I trained models with three-class output labels.

Prediction with 1 & 8 of Market Data as Input



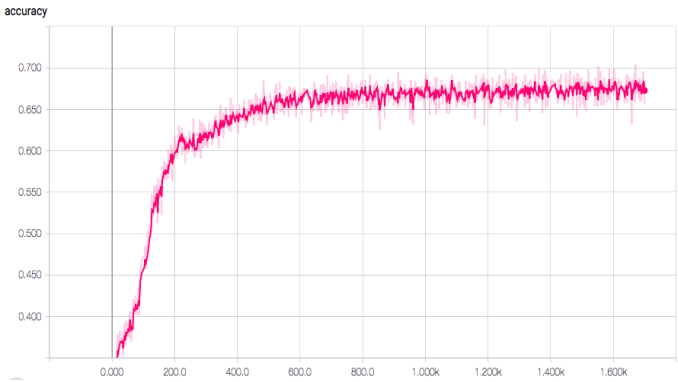


3. Predicting Price Trends

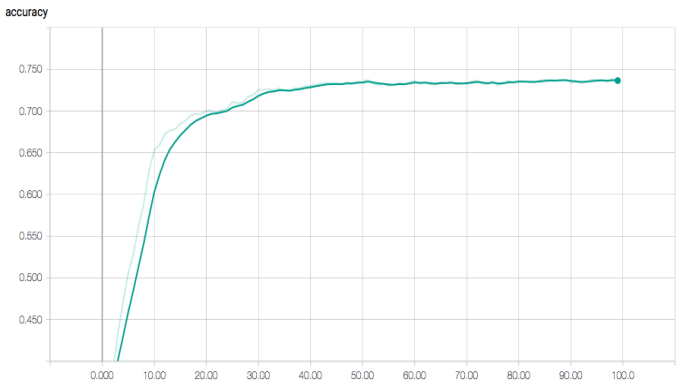
Specifically, I categorized the next period's returns into three types, used market data of last 30 days to predict the price trends in the next period, and the output labels were one-hot variables "100, 010, 001" respectively representing "rise, moderate fluctuation, decline", determined by 1/3 quantile (this meant that original accuracy was 0.3333).

As for long-term prediction in price trends, the out-of-sample test accuracy of RNN finally reached 69.4% and the accuracy of the optimal LSTM model achieved 73%.

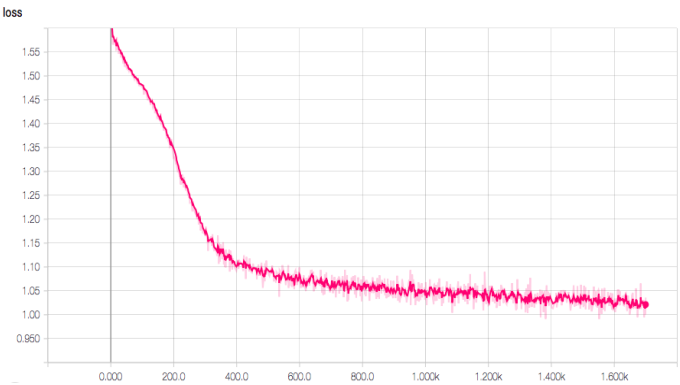
Accuracy Convergence Curve in Training Set



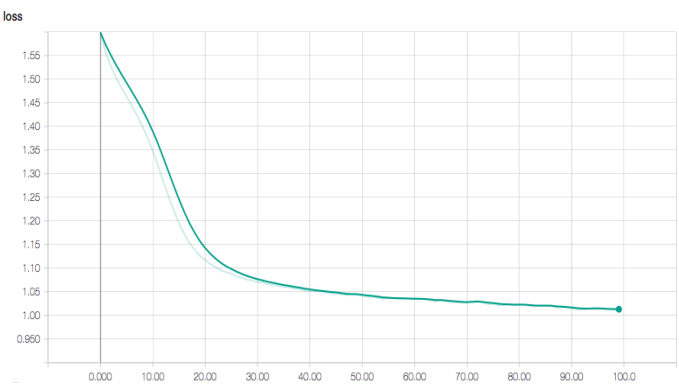
Accuracy Convergence Curve in Test Set



Loss Convergence Curve in Training Set



Loss Convergence Curve in Test Set



IV. Conclusion

In the research, I took 8 types of market data of CSI300 Index constituent shares from Jan.1st 2006 to Jul.31st 2018 as samples, constructed RNN and LSTM models to predict the next rolling period's returns / price trends respectively. My experiment showed that more input information yielded a more accurate prediction and the best model achieved an out-of-sample accuracy of 73% in long-term three-class prediction for price trends.