Deep Learning Application in Stock Market

Shuning Huo
Virginia Tech, Department of Statistics
shuni93@vt.edu

Abstract

Deep neural network models have brought recent advances in computer vision and natural language processing. The hierarchical models implement unit to represent hidden patterns behind data, a non-linear mapping of complex data structure. Financial data is a source of very high-frequency, complex signal data compared to other ones. Instead of finding the online covariance matrices in traditional way, deep learning has provided us alternatives to express financial data. We would like to apply deep learning models in stock market to see if they have some common hidden patterns so that we are able to utilize. We would like to analyze whether a stock will have certain increase in certain period after some data preprocessing work. And if we can predict it will rise or not, we come to the question that when it will achieve expected return. In this report, we are going to implement CNN, RNN(LSTM) models to classify the financial data.

1 Introduction

Deep neural network model has attracted tremendous attention recently due to their accuracy in difficult tasks, such as image classification, object segmentation, natural language process.

Financial prediction problems are of great practical and theoretical interest. They are also quite daunting. Theory suggests that much information relevant to financial prediction problems may be spread throughout available economic and other data, an idea that also gains support from the many disparate data sources that different market participants watch for clues on future price movements.

In this report, we apply deep learning hierarchical decision models for problems in financial prediction and classification. The deep learning predictor has a number of advantages over traditional predictors, which include that:

1.input data can be expanded to include all items of possible relevance to the prediction problem, such as images, text,

2.non-linearities and complex interactions among input data are accounted for, which can help increase prediction capability,

3.over-fitting is more easily avoided.

The report will use image data, sequence financial data to classify and predict financial data. Section 2 will have a general look at the widely used deep learning model, CNN and LSTM. Section 3 will introduce our experiment for training process.

2 Technical Approach and Models

2.1 Convolutional Neural Network

Convolutional Neural network model has become a powerful way to classify image data. It applies convolutional layer and max pooling techniques to downsize sample size while extracting the features of images. The core theory behind CNN model is using multi-depth filter kernel to extract local features of each image. The following figure shows details of convolutional layer, which filters out $32 \times 32 \times 3$ images.

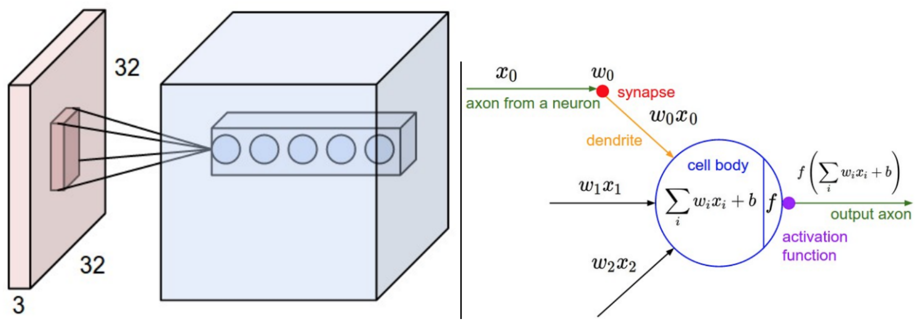

Figure 1: CNN layer

In the next section, we will train our model using multiply layer convolutional neural network to classify stocks.

2.2 LSTM

In natural language processing, RNN (recurrent neural network) has played an important role recently. Specifically, neural network assumes that our input features are independent with each other, which may become a simple but unreasonable assumption. Sentences, paragraphs, video and other data reveal sequential characteristics which need to be taken into consideration when modeling. RNN model provides us a good option to deal with such time relevant problems. Compared to other neural network model, RNN takes advantage of a specific unit called gate to store the previous information of input unit. However, RNN model only restores short-time information. LSTM, which is a revised RNN model, is able to memorize long-term information. Also, it has forget term to decide if the information will be retained. We will apply LSTM model in our financial data to track sequential traits. The basic idea behind LSTM unit is introduced in the figure 1.

$$
\begin{aligned}
\mathrm{z}_t &= \sigma(W_z[h_{t-1}, x_t]) \\
\mathrm{r}_t &= \sigma(W_r[h_{t-1}, x_t]) \\
\tilde{h}_t &= tanh(W[r_t * h_{t-1}, x_t]) \\
\mathrm{h}_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned}
$$

where z_t, r_t, $\tilde{h}_t$, h_t denotes the update gate, reset gate, reset memory and new memory respectively.

3 Experiment

3.1 Data preprocessing

Data preprocessing is very important in our task since financial data is of high frequency and high noise. We set up our own potential database in this step by using small window, finding 90-days period which contains 5 consecutive days growth. The procedure is shown in the following steps:

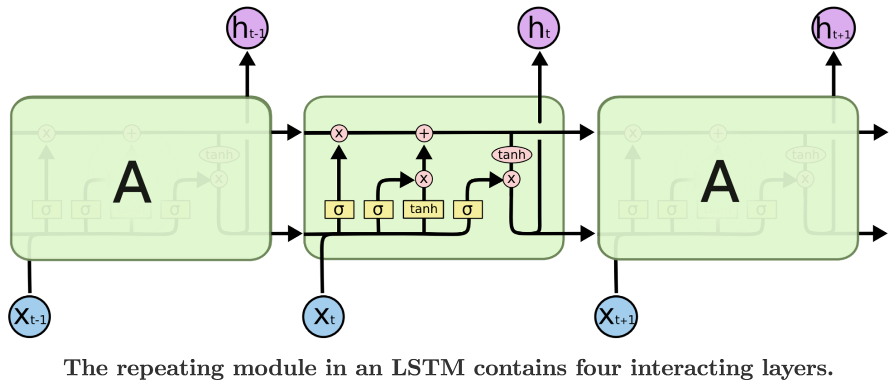

Figure 2: LSTM unit[9]

- Retrieve SP500 daily close price, volume from Internet from January 3 2000 to March 26 2018.
- Use the small window to filter every stock to find 90 period days with 5-consecutive-days growth and set up entry point for every 90-period days.
- For every 90-day period, looking forward to continous 90-days period to see if it increases with 10%, decreases 10% compared to entry point or nothing happens(censoring). We denote them as +1, -1, 0 respectively and store the days when they achieve the targets if it is not 0.
- Generate rate of return plots based on our 90-days stock data by using the entry point as the baseline and calculating return.
- Store either the plot matrix or sequence stock data in our database.

We collected about 22826 training examples in database. The images we generates are just gray-scale images like mnist and we decided to use 128×128 pixels. The following figures show two examples of our preprocessed data.

Figure 3: Example return plot.

Table 1: CNN architecture

Layer	Filter size, stride, number
Conv(Relu)	3×3, 1, 32
Conv(Relu)	3×3, 1, 32
Max-pooling	2×2
Conv(Relu)	5×5, 1, 64
Conv(Relu)	5×5, 1, 64
Max-pooling	2×2
FC	512
FC(Softmax)	3

Table 2: LSTM results

LSTM	LSTM (with normalized daily volume)
78.4%	77.88%

3.2 Convolutional neural network

The first model we tried is convolutional neural network which has reliable performance in many computer vision tasks. Based on our created database, we will have classification about whether the stock will increase by 10%, decrease by 10% or censored. The specific CNN model we tried has the structure shown in Table 1. The model can be easily implemented in keras library, as provided in code.

Our test accuracy is about 62.5% in this scenario. Since the training process takes a lot of computational resources and time, we only tried 8 epoches. However, I believe there can be much improvement by tuning parameters.

3.3 LSTM

Different from what we have tried in last session, we will apply LSTM model to make classification by daily return instead of using generated return images. Financial data are inherently time-series data and we believe it will be helpful for us capture the traits of it. In addition, day-trading volume plays an important role in stock pricing so we can incorporate useful financial information into classification task.

The first model we tried is simple LSTM model daily return data. We used 1-LSTM-layer with 128 hidden units. We take every sequence of return data as input so we have 90 time steps. The specific implementation can be seen in the code materials. The classification accuracy is about 78.4%, 15% higher than that of CNN .

Furthermore, we incorporated normalized daily volume as feature of LSTM model input and see whether such information can provide us better results. Keeping other hyper-parameters same with the previous one, we are able to achieve 77.88% in accuracy. We can not see much improvement by incorporating volume into the model. Or there may be more feature engineering work to be done.

Compared to CNN model tried in previous section, LSTM has better performance in our financial data application. Besides, it is much less computational intensive which gives us more flexibility.

3.4 Survival Analysis

Based on the training results of previous deep learning models we can say that there are some patterns behind stock market for us to classify. The next question to be answered is when is the day such a stock will achieve the target? Since we care about whether it will increase or decrease within some period, we will analyze samples with 1 or -1 classification results. Without losing generosity, we only analyze those with 10% increase stocks.

We Extract those prediction stock data as 1(10% increase) and use LSTM output layer(sigmoid activation function output) as our feature to fit cox hazard proportional model. Also, we deleted the first output of sigmoid function since summation of three output is equal to 1.

The two features we generated from LSTM model are both significant in predicting survival hazard function, as shown in the following plot. We can also provide hazard prediction if we would like to look into in which period the stock will increase 10%.

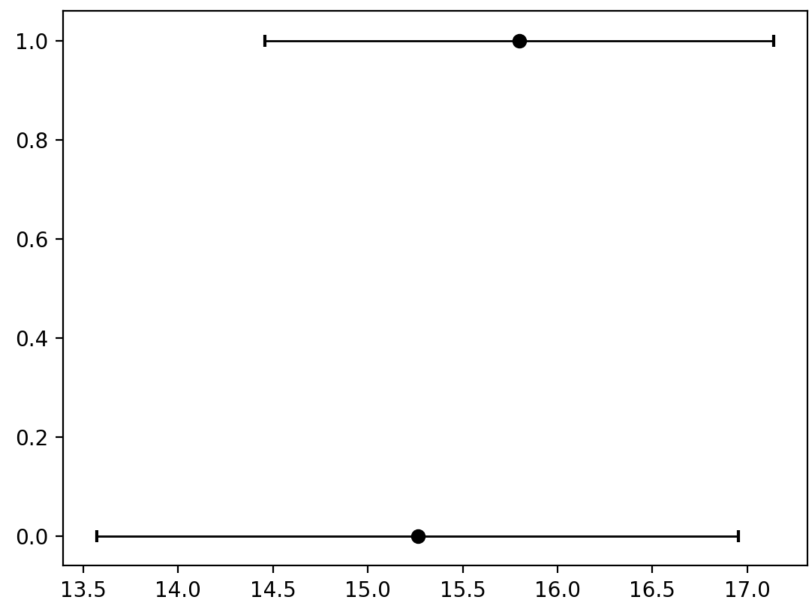

Figure 4: Cox hazard coefficients

4 Conclusion and Further Discussion

Deep learning presents a general framework for using large data sets to classification and prediction. As such, deep learning frameworks are well-suited to many problems both practical and theoretical in finance. The application presents deep learning hierarchical decision models for problems in long-term stock prediction and classification. We also notice that LSTM outperforms CNN in a large scale due to time-series traits of financial data. Furthermore, generated images are sparsely represented by 0(black pixel) and 1(white pixel), leading to worse classification results. Deep learning can also be a tool for us to generate features for further study. There is still much work can be done by utilizing deep learning architecture in finance industry. Further work can be focused on develop specified loss function to deal with financial time-to-event data.