Stock Prediction with Convolutional Neural Network Fed with Images

**Abstract**

Prediction of stock market attracts researchers from many fields, like finance, economics, history, mathematics, and computer science. Recent years, machine learning is the main method for researchers from computer science to forecast stock price. However, most of their approaches use 1D vectors to feed models. In this paper, a model based on deep convolutional neural network (CNN) is fed with images of history data. Two CNN networks, Alexnet and VGG19, are implemented, and VGG19 has a better result with best accuracy about 58%.

.

## Introduction

**Stock Prediction**

When an individual buys a stock, he essentially becomes one of the owner of the company which sells the stock. The stock market is a community in which some people want to buy stocks while someone wants to sell. In finance and economics, there is a law of demand and supply which holds that when the demand increases or the supply decreases the price would increases. On the contract, if the demand decrease or supply increases the price will decreases. Stock also likes these common commodities to some extent in the stock market. The stock market is also a well-organized market where people can sell and buy in a public manner.

Stock prediction has been an intriguing topic for a long time. In general, people would buy the stock before the price rising and sell it before drop. Stock prediction is to predict the stock trend or the stock price, which can help stock buyer to know when to buy and sell, to make more profit. In general a few percentage points of stock prediction improvement can increase profit by millions for those investors. Or help the stock holder against systematic risk. Because the stock market is subject to large price volatility. For common shares holders this large volatility means high risks. That is why the stock prediction is significant important to against systematic risk.

Unfortunately, stock prediction is not simple as it said in the law of demand and supply. Stock market is essentially dynamic, non-linear, nonparametric and chaotic in nature. Beside the demand and supply, it is also influenced by many macro-economic factors such as political events, company’s policies, general economic conditions, commodity price index, bank rates, investors’ expectations, institutional investors’ choices and psychological factors of investors. [1] In addition, according to Eugene Fama (1970)’s Efficient Markets Hypothesis (EMH), the stock always trade at a fair value on stock exchanges, making it impossible for investors to either purchase undervalued stocks or sell stocks for inflated prices. As such, it should be impossible to outperform the overall market through expert stock selection or market timing, and the only way an investor can possibly obtain higher returns is by purchasing riskier investments. The current price essentially satisfy the law of demand and supply and the prices movement is like random walking (RW) (Robert Brown). But the EMH is based on a questionable presumption that the investors in the market are sufficiently rational and can respond promptly to all market information. And in recent years many stock prediction methods has already achieved better results than the RW. Therefore it is possible to predict the stock price to some extent. Overall, predicting stock price movement accurately is not only interesting but also extremely challenging.

**Stock Prediction with Machine Learning**

As said before, many researchers of computer science also did a lot research in stock prediction. The main method is machine learning is a well-established method which is in a wide range of applications. Machine learning is the science of getting computers to take decisions without being explicitly programmed to do so. It can give satisfied results in some areas like image recognition, text recognition and speech recognition. It can be used for almost anything. With the help of machine learning, a lot of complex problems can be solved as long as there are enough data and computation resources. While the improving of the computing ability of computers in 20th century, machine learning has been an important method of solving complex problem. It can be divided into supervised, unsupervised, association and reinforcement learning. In supervised learning, input data will be labeled to represent the class of the data. This kind of learning algorithm includes Decision Tree, Regression, Support Vector Machines (SVM) and Naive Bayes. While the unsupervised learning algorithms use unlabeled data to draw inferences from beforehand. These algorithms include K-mans and Gaussian Mixture Models.

In machine learning area, SVM and Artificial Neural Network (ANN) are mostly used for forecasting stock price. SVM was first provided by Vladimir N. and Alexey Ya Chervonenkis in 1963 and has become one of the most popular algorithms in machine learning with half century development. While ANN possess attributes of learning, generalizing, parallel processing and error endurance, which makes ANN powerful in solving complex problems like stock prediction (Azadeh, Ghaderi 2008)

## Related Work

Previous works about stock forecasting can be divided into two parts according to input data, including stock prediction with history data and stock prediction with news events.

**Feed with history data**

Most of the methods are feed with history data. History data are the price history collected by some financial company like Yahoo Finance and Google Finance. These data mainly contain 6 columns, time, open price, close price, high price, low price and volume. The opening price and close price is the price at which a security first trades and the last trades upon the opening of an exchange on a given trading day, while high and low represent the highest and lowest price among the time period. Since the stock prediction is one of the time-series-forecasting problem, many researchers use the history price with time to predict the stock movement. Also, according to machine learning method, these methods mainly include two parts, SVM and ANN.

**ANN**

Most researches with machine learning prediction has focused on ANN, which is constructed with a series of interconnected nodes. This model simulates individual neurons which are organized into different layers based on function. The model assigns weights to connections and calculate the outputs with the weights. Preminger and Franck [2] used a robust linear autoregressive and a robust neural network model predict stock, but this model is better than the RW. Yu and Huarng [3] used a model called bivariate neural network-based fuzzy time series with substitutes to predict time series problem. Then Change et al [4] provided a method called CBDWNN which combines dynamic time windows, case based reasoning and neural network with other methods. They compared that method with some other main methods and CBDWNN had a better results.

In 2011, Erkam Guresen et al [5] combined the ANN with genetic algorithm to build a new method for stock prediction. They use polynomials to build ANN, then use genetic algorithm to estimate parameters of ANN, like starting polynomials and weights. This model is a regression model. However, the result of this model is not satisfied, but it is a good begin of combination of ANN with other optimal algorithms. In the same year, Jianzhou Wang et al [1] provided an approach which can remove the noises in stock data to some extent. The method decomposed data into multiple layers by wavelet transform (WD). With this, input data would be divided into high-frequency and low-frequency data. Then establish back propagation neural network (BPNN) with those low-frequency data. One thing need mention is that they only use is close price of history data. The finial result is slightly better than normal back propagation neural network.

Kar [6] implement a RNN model to predict stock market. In this model a number of activation functions are implemented along with options for cross validation sets. The input data contains open, close, high and low price of history data. This model is a regression model and output is predicted close price. Average accuracy is 88%, seems much better than other works, but this accuracy is not the like other work. The accuracy here means the difference between predicted close price and actual close price. As price value sometime is always more than thousands, therefore it is good but not good as it looks like.

**SVM**

SVM is powerful in tracing the stock market and helping maximize the profit of stock option purchase while having low risk. Compared with ANN May finding a local optimum, SVM has more chance to find a global optimum. Therefore, numbers of researchers have tried to using SVM to predict stock price. Kim’s [7] method based on SVM can predict the stock price with a 57% accuracy which is significantly above 50% threshold. Shah [8] has compared some machine learning methods for stock prediction. Among those methods SVM has achieved the best accuracy which is about 60%.

Instead of national stock data, Zhang [9] used global stock data in associate with data of other financial product as input features to SVM. Beside history data of stock, they also uses some related financial products, like currency price (USD, AUD, EUR and JPY), resource price (silver, platinum and oil). He compared the prediction accuracy of single features. As expected the currency price has significant influence on the prediction result. Also, he found the best term of prediction is 20-30 days.

Dai [10] also compared the svm with other methods on the history data of Minnesota Mining and Manufacturing (3M). SVM had a 55.2% accuracy.

Madge [11] trained a SVM model with 34 technology index. In this wok, he predicted the price volatility and momentum for individual stock and overall sector. He calculated the average price changing rate of ‘n’ days to represent the price volatility while using average label of n days to represent the momentum. When the price of next is higher than today, it is labeled as ‘1’ otherwise ‘-1’. The prediction term ‘m’ is better to be between 20 and90. While increasing ‘m’, the model need more data from this stock, and less form other stocks. This would lead to a higher accuracy on this particular stock but much lower on other stocks. Therefore, prediction term should be between 20 and 30 to keep a good accuracy on all the stocks.

**Feed with news events**

Different with those approaches feed with history data, some approaches feed with news events also has been provided in recent years. In the most recent year, Ding [12] found a method to use news events to predict stock price with CNN in 2015. First, useful information would be extracted from 10 million finance news with Natural Language Processing (NLP). Second, a CNN model would be built with these vectors to predict price trend. With this model he can achieve a best accuracy of 65%, which is a great improvement for stock prediction.

In the next year, Poulos [13] used another RNN model to predict stock price. Similarly, he uses NlP for extracting information from news, but difference is that Poulos uses news titles. And his RNN model is based on Gated Recurrent Unit (GRU). Totally, this model has 12 GRU with Dropout layers, then ending with 1 dense. With same datasets, the result is 55.94, slightly better than 3-layer LSTM (55.45%) but worse than SVM (56.31%)

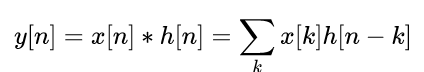
## Stock Prediction with Images

Now it can been seen that, most of the approaches are fed with 1D vectors, no matter history data or news events. For a stock investor, he would look at the images of stock price for more time than the history data. And for those who want to buy shares, pictures like candlestick chart is more convenient to obtain and easier to understand. Therefore, in this project, a model fed with images about price information would be implemented with machine learning. As it is an image classification model essentially, Convolutional Neural Network (CNN) would be used since its power on image classification problem.

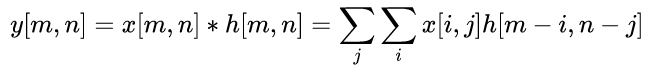
**Convolutional Neural Network**

Convolutional Neural Network (CNN) was proposed in 1960s by Hubei and Wiesel when they were studying the neurons of local sensitive and directional selection in the cortical cortex. They found that their unique network structure can effectively reduce the complexity of the feedback neural network. Inspired by this, they proposed CNN. Now CNN has been developed in recent years and become a high efficiency identification method. CNN has also become one of the hotspots in many fields of science, especially in image classification.

For 1D vector, convolution can be represented as follow:



x[n] is input signal, h[n] is the unit response while Y[n] is output signal. For 2D signal, the convolution is:



Therefore, for image processing, convolution is to find the weighted sum for the pixels of the image with given convolution kernel. The main aim of convolution processing is to reduce space cost with time cost. In general, the more convolution kernels given, the more feature maps would be extracted and the results should be more accurate, but more parameters would needed.

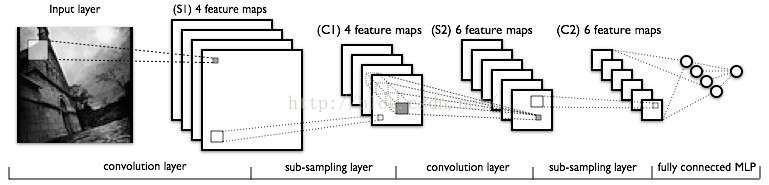


Figure 1. Convolution for images

**Pooling Layer**

In the pooling layer, a value would be used to represent an area. This value can be the average, maximum, minimum value of the area of some other values. The main reason to add this layer is to reduce the parameters.

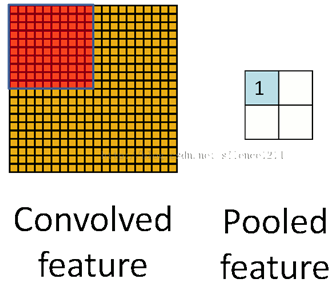


Figure 2. Pooling layer

**Activation layer**

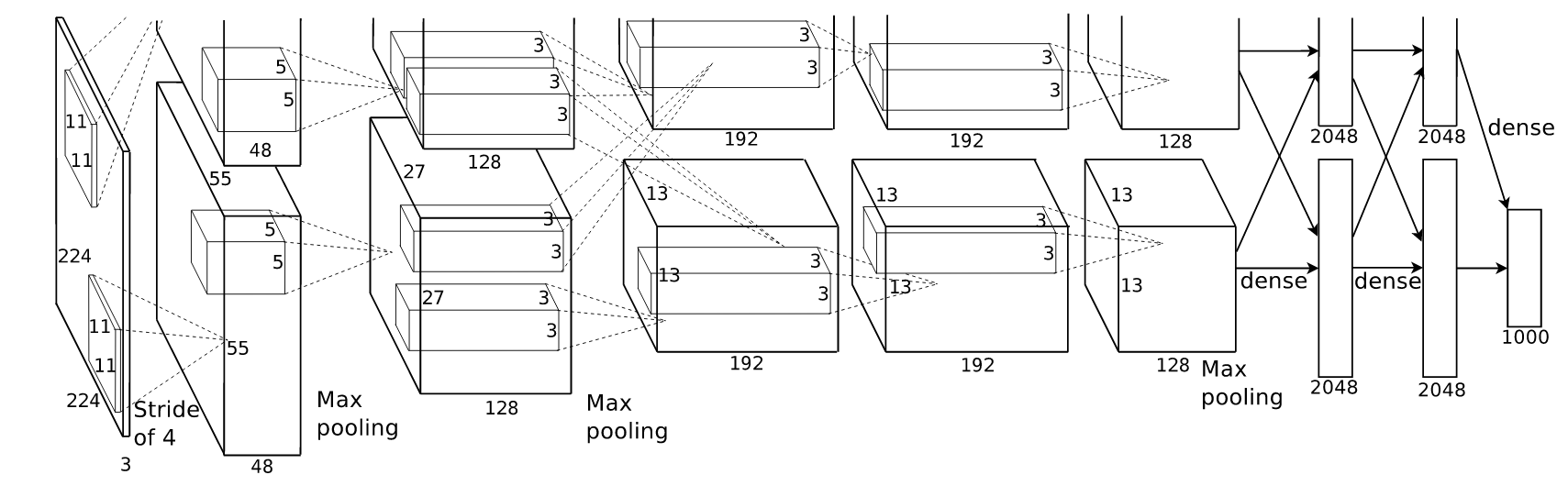
Activation functions are used to ensure that the convolved results are in a fixed range. This can keep the value range is controllable after several times convolution. Some common activation functions include sigmoid, tanh, ReLu. The value range of them are [0, 1], [-1, 1] and [0, Infinity]

### CNN model

In this paper, two models are used, including Alexnet and VGG19.

**Alexnet**

This network is provided by Alex [14] on the imagenet image classification challenge, which won the champion in 2012. This is a classic model of image classification. Therefore, this model is chosen to do the first attempt for stock prediction with images.

 Figure 3. Alexnet network structure [14].

As showed in Fig.3, Alexnet has 5 convolutional layers, 3 fully connected layers and a softmax at the end. The activation function used in this model is ReLU. Kernel sizes are 11, 5 and 3. Fully connected layers’ sizes are 4096, 4096 and 1000.

The feature map size is calculated by:

new\_feature\_size = (Img\_size–filter\_size+2\*padding\_size)/Stride+ 1

For example, for a 224\*224 input image, it will be per-processed to 227\*227. Then the feature map size after first convolution layer is (227-11)/4+1 = 55. The output number is 96. Therefore, there will be 96 feature maps with size 55\*55. The reason why it is 48 in Fig.3 is that they are divided into 2 parts when using 2 GPU. During the first pooling processing, the output is 96 feature with size (55-3)/2+1 = 27. Similarly in the second convolution layer, the feature size is (27-5+2\*2)/1+1 = 27 and the output number is 256. After the last convolution layer and pooling, the output size is 6\*6, output number is 256. These neurons would be connected with 4096 neurons in next fully connected laysers.

In the first convolution layer, there are 95 output feature maps. Each of them has 11\*11\*3 (kernal\_size\*channel) weights and 1 bias. Therefore in this layer, there are 96\*363 + 96= 37824, and 60M parameters in total.

Similarly the total parameters of Alexnet is about 60M

**VGG19**

Then another deeper network called VGG19 [15] is also be implemented. Compared with Alexnet, VGG19 has 16 Conventional layers which is much deeper than Alexnet, but similarly they both use 3 fully connected layers. Another difference is that VGG19 uses fixed kernel size (3\*3). They both use same activation ReLU.

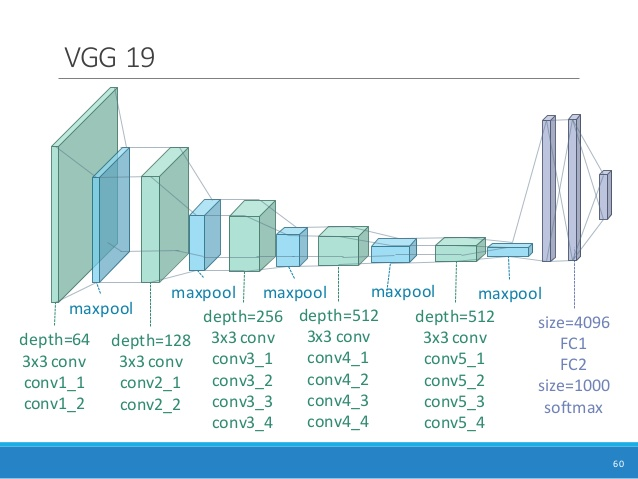


Figure 4. VGG19 network ([Mark Chang](https://www.slideshare.net/ckmarkohchang?utm_campaign=profiletracking&utm_medium=sssite&utm_source=ssslideview))

The output size and output number can be calculated similarly as introduced in Alexnet. The output number of each layer is much bigger than previous model. As expected, the parameters of this model is much more than Alexnet. The total parameters of VGG19 is about 144 millions.

## Experiment

**Data Collection**

In this experiment, images represented history data would be used as input. First, history data need to be collected. 3000 stock indexes are randomly chosen from Yahoo Finance. Then history data of them in 15 years before 2017 were downloaded. Unfortunately, because of data loss or network issues, only 2397 indexes were downloaded. Then during per-processing, about 400 indexes were removed because of some ‘NAN’ value. Before removing them, the closed value had been used to represent these ‘NAN’ value, but the results is unsatisfied. Therefore, all these datasets with ‘NAN’ values were removed from the datasets.

Open, close, high and low price are used in the experiment. At first only high and low price were used, since high and low contains more information according to Siripurapu[16]. High and low price are the bound of daily stock price, which also contain the open and close price. Open and close is in a sense of statistical artifacts, which are the prices sampled by the Google or Yahoo. However, after tried with only open and close price, the result was not good. Information on input images only with open and close price is not enough for deep network to extract enough useful features. Therefore, in next experiment, four kinds of price of stock are input into the network, then the network can automatically choose which one is more useful.

Then images of stock price would be poled as RGB with size 5\*5 kb. The windows size of image is n {180, 90, 60, 30}, which means using n days history data pictures to predict future m {30, 20, 10, 5} days. An example input image is showed as follow:

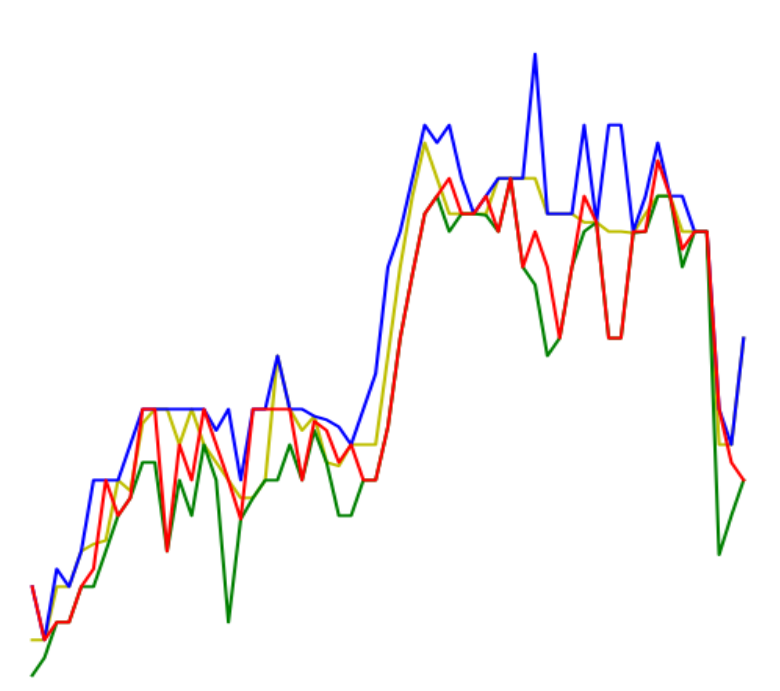


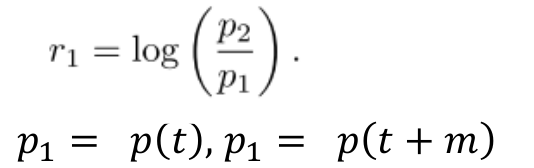
Figure 5. Input example of n = 60

Then this picture would be stored into Tfrecord and its size would be resized as 224\*224.

**Prediction strategy**

As said previous, prediction strategy is using previous n day’s history data pictures to predict future m days. 4 combination of n and m are used including using 180 days history data to predict 30 days price, using 90 days history to predict 20 days price, using 60 days history data to predict 10 days price and using 30 days history data to predict 5 days price.

These pictures would be labeled by the return of the price after m days. The return here is logarithmic return.



Price is mean of open, close, high and low price. ‘t’ is current time and ‘t+m’ is the future price. When ‘r’ > 0, it will be labeled ‘1’ which means ‘up’, otherwise it is ‘-1’.

The image numbers for training and testing are always about 8:2. The image number also increases with the increasing of model’s depth. When using the Alexnet, the image number is from 1000 to 10000, while using at least 50000 to feeding the VGG19 model.

Cross entropy is chosen as loss function and optimal is Gradient Descent Optimal (SGD). Learning rate is from 0.01 to 0.00025 while Dropout rate is 0.5.

### Results

After compared the combination of n and m, I found prediction term m should be 5 or 10 days, while history data used to predict should be 30 or 60 days. Using 60 days to predict 10 days has the higher accuracy in this experiment. Then followed experiment is based on 60 days to predict 10 days.

With 8 layers model, training an Alexnet model need only a few time which is less than 2 hours. This model converge quickly even with a learning rate of 0.01. However, no matter how to adjust the parameters or change input data (like change n and m, or use different datasets), the model is always overfitting. The accuracy of this model is close to the RW, which is around 50% and the loss is always more than 0.69 as showed in Fig 6. However, when I try to increase the training data number, the gradient disappears quickly even with the smallest learning rate 0.00025.

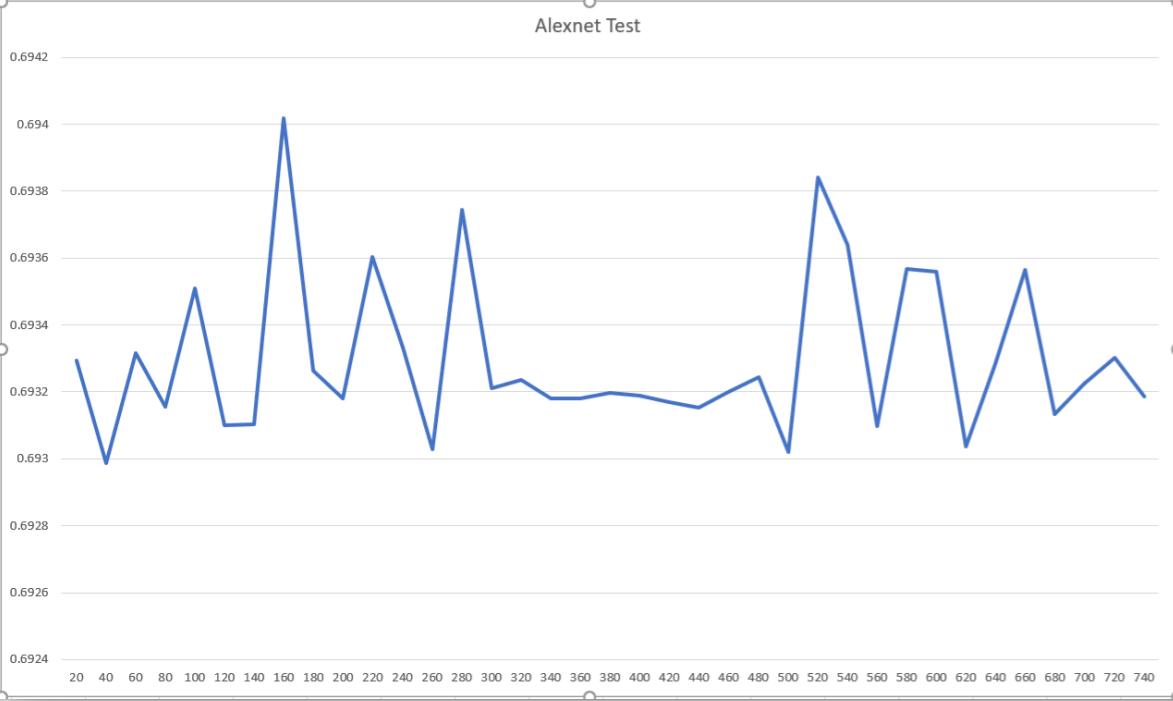


Figure 6. Test loss of Alexnet

As expected, with much more layers network, VGG19 need much more time to train. Because this mode takes more memory, I had to reduce the batch size to 15 from 200, or it will out of memory on an Nvidia 3G GPU (GTX 970M). It takes about 5-6 days to train a new model. However, this model does have a much better results than the Alexnet. As showed In Fig.7 it can achieve a smallest testing error about 0.61 and with the best test accuracy about 58%. Unfortunately, with the training goes on, the test loss would increase again, since this model are also overfitting.

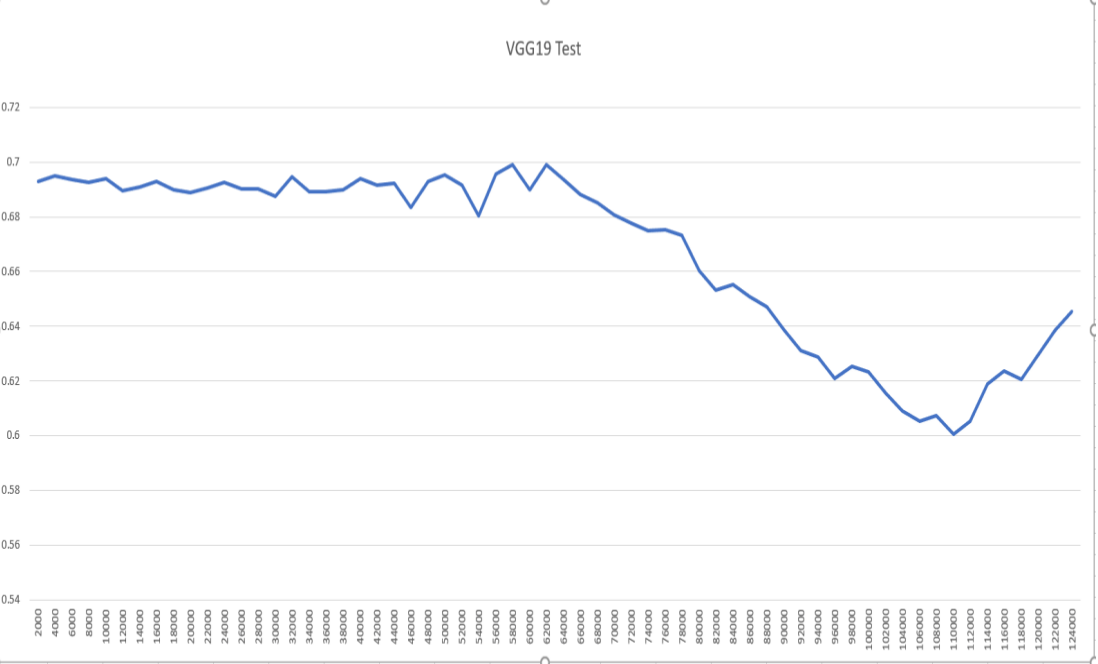


Figure 7. Test loss of VGG19.

### Discussion

There still exists a lot of aspects need to be improved. First is about the data collection. 100,000 images for tanning is still not enough, and 1 million pictures are needed for training. To adjust this, another more powerful GPU is needed in the future work. History data used in this paper are from 3000 randomly picked stock indexes. That’s may be one issue. As showed in other’s work, most of them uses similar stocks or single stock. Therefore, in the future work, stocks in same sector, like technology sector, would be used for training and testing. Because in the same sector, stocks have more similar features, which can help the model to predict easier.

Also, beside close, open, high and low price, more feature values should be plotted into the pictures, like average price of all indexes. Currency price is also another significant feature, like AUD, USD, EUR and others. Currency price represent the economic policy to some extent, while economic policy is a great influence on stock price.

To avoid all lines being entangled with each other, channels would represented different lines instead of RGB. Then input pictures would be convert into gray images. Therefore the input of model would be 224\*224\*5. 5 channels means open, close, high, low and average price. This can be more if currency price is also added in future work.

### Conclusion

In conclusion, this paper represents a method of using images to predict stock price. The reason for choosing images as input instead of 1D vectors is that images about stock price like candlestick chart are more often used for stock investors and easier to understand. Compared with feeding with 1D vector, this approach almost does not need any per-processing, and the model can feed with raw pixels. In addition, images can contain more than one time-series. Another reason is that convolutional network is more stable. However, the brought disadvantage is obvious that this model need much more time to train with bigger input. Two networks are used in this paper, one of them is Alexnet and another one with much more layers is VGG19. As expected VGG19 is much better than Alexnet in this problem, which has a best accuracy about 58%, while taking much more time for training.

**Reference List**

[1] J. Z. Wang, J. J. Wang, Z. G. Zhang and S. P. Guo, forecasting stock indices with back propagation neural network, In Expert Systems with Applications, Vol. 38, pp. 14346-14355, 2011.

[2] A. Preminger and R. Franck, R. Forecasting exchange rates: A robust regression approach, In International Journal of Forecasting, Vol. 23, pp. 71–84, 2007.

[3] T. H. Yu and K. Huarng, A bivariate fuzzy time series model to forecast the TAIEX. In Expert Systems with Applications, Vol. 34, pp. 2945–2952, 2008.

[4] P. C. Chang, C. H. Liu, J. L. Lin, C. Y. Fan, and C. S. P. Ng, A neural network with a case based dynamic window for stock trading prediction In Expert Systems with Applications, Vol. 36, pp. 6889–6898.

[5] E. Guresen, G. Kayakutlu and T. U. Daim, Using artificial neural network models in stock market index prediction, In Expert Systems with Applications, Vol 38. P 10389-10397, 2011.

[6] Abhishek Kar (Y8021), Stock Prediction using Artificial Neural Networks.

 [7] K. J. Kim, Financial time series forecasting using support vector machines, In Neurocomputing, vol. 55, pp. 307-319, 2003.

[8]V. H. Shah, Machine Learning Techniques for Stock Prediction, pp. 1-19.

[9] T. D. Zhang, S. R. Shen and H. M. Jiang, Stock Market Forecasting Using Machine Learning Algorithms, pp. 1-5, 2012.

[10] Y. Q. Dai, Machine learning in Stock Price Trend Forecasting, pp. 1-5, 2013.

[11] S. Madge, Predicting Stock Price Direction using Support Vector Machines, In Independent Work Report Spring, pp. 1-10, 2015.

[12] X. Ding,Y. Zhang and T. Liu, Deep Learning for Event-Driven Stock Prediction, In Twenty-Fourth International Joint Conference on Artificial Intelligence, pp.2327-2333, 2015.

[13] J. Poulos, Predicting Stock Market Movement with Deep RNNs, pp. 1-5, 2016.

[14] A. Krizhevsky, I. Sutskever and G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, 2012.

[15] K. Simonyan and A. Zisserman, Very Deep Constitutional Network for Larger Scale Image Recognition, Visual Geometry Group, Department of Engineering Science, University of Oxford, 2014.

[16] A. Siripurapu, Convolutional Networks for Stock Trading, pp. 1-6, 2016.