Stock Price Prediction by Deep Learning

Arnat Leemakdej

E-mail: ARNAT@TU.AC.TH

Thammasat Business School

Abstract

The study investigates how the computer can convert unstructured data into word vectors and use them to predict stock price. First, the machine learns words from financial news and transform them into vectors. These contextual vectors are fed to TensorFlow's deep learning system to predict future prices of key stocks listed in the Stock Exchange of Thailand. The results suggest that there is still a room for investor to gain abnormal return by trading within 5 minutes after news release.

I. Introduction

The efficient market hypothesis (EMH) is one of the main pillars of modern finance theory. Fama (1970) synthesizes empirical studies on stock price prediction and concludes that the stock exchange is efficient if all information immediately reflected in the stock price. However, he admits that there are three level of informational efficiency. Beginning from the weak form efficiency, historical data cannot be used to predict future price. This rejects the ability of both technical and fundamental analyses to make abnormal profit in the market. The more stringent level is semi-strong form efficiency that once the data becomes public, investors cannot benefit from such data. Investors who closely follow news cannot make profit once such a news is announced. The strictest one is the case of strong form market such that even insiders cannot gain from trading by an unannounced news.

Many studies find supports in market efficiency at the level of weak form and semi-strong form. This paper argues that the tests of semi-strong form in previous studies are usually conducted on a daily data basis. The conventional test is an event studies proposed by Fama et al (1969). Closing price is used to measured abnormal return on the day that the news is released. Many investors, in particular day traders, do not hold position until the end of the day. They can buy and sell in short interval after news announcement. This paper will test if an investor can make abnormal return from public information by trading within a short period after news release.

This paper contributes to the knowledge of finance and computing in three areas. First, I reinvestigate the semi-strong form efficiency of the market. Public information in this study is not limited to only the corporate event of the firm but include all news publicized. The test is conducted in a short interval after the announcement. Second, I contributes to the analytics of a big data. Thai news in unstructured format are converted into word vector by means of machine learning. Third, I use deep learning tool to predict the stock price movement from public news. The study sheds light on the application of big data and sophisticated machine learning to modern financial theory.

Predicting the ten most liquid stocks traded in the Stock Exchange of Thailand by the deep learning, this study shows that the market is not semi-strong form efficient. The 5-minute trading strategy after news released can yield a significant 18.2 percent daily return on an average.

II. Literature Review

A number of researchers has been investigating how to use financial news to predict stock or commodity price recently. This is due to a vast amount of data becomes public and new advanced methods have been developed to deal with big data.

I classify recent literature that use financial news to predict the future into two groups. The first group [e.g. Bollen et al (2011), Feldman et al (2011), Kim et al (2014), Nguyen et al (2014), Chattupan and Netisopakul (2015), Joshi et al (2016), Heston and Sinha (2016)] is called sentiment approach. This group uses simple statistical technique to count the frequency of positive words and negative words appeared in news, then, calculates the sentiment score. The relationship between stock price movement and sentiment can be simply investigated by a regression model. The second group, in which this study also belongs to, [Enke and Thanomwong (2005), Schumaker and Chen (2009), Luss and d'Aspremont (2009), Fai et al (2014), Ding et al (2015)] uses a neural network to convert unstructured data into vector. This group proposes that words in a particular knowledge domain share commonalities which can be extracted by the nodes in hidden layers of a neural network. The neural network is used again to model the relationship of news vector and future stock price. Fast computing power now allows researchers to add hundreds hidden layers in the model. They now refer to the sophisticated neural network as the deep learning.

All studies but one, Luss and d'Aspremont (2009), support that the stock price prediction can be improved by using either the sentiment or the deep learning approach. Only few of them [Schumaker abd Chen (2009), and Luss and d'Aspremont (2009)] use intraday data to test trading strategy.

III. Data and Methodology

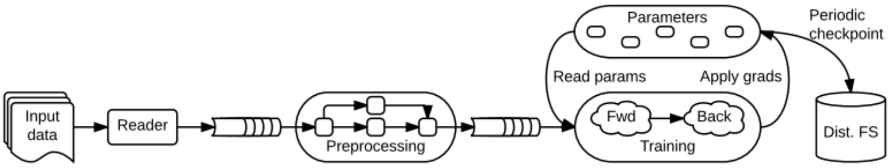


Figure 1 The Study Process
Source: Google Brain Team (2016)

Figure 1 illustrates the processes used in this study. First, the news corpus were extracted from NewsCenter. Words in sentences were separated by Thai dictionary. They were then converted into word vectors as suggested by Mikolov et al (2013). Word vector is a mathematical concept that uses surrounding words to predict the target word. There are two methods to predict, namely the continuous bags of words (CBOW) and skip-gram. This study uses the latter approach with 5 words before and after the target word. I define 100-dimension vector to describe one word. This study extracts 78,995 news during March 3, 2016 to August 31, 2016. There are 99,469 words found in this study.

To limit the complexity of the network, only news header is used to describe news. The number of words in news header ranges from one to forty-nine with the average of twelve and the 75th percentile at fifteen. I select fifteen words, each with 100 dimensions, from the header as the input for the deep learning in the next step.

The deep learning process involves key decisions on what will be the inputs, how many hidden layers needed, how each node is inter-connected, how to reduce dimensionality, how to define output. These concepts are briefly explained below.

The simplest form of neural network has input layer; in our case this is the word vector produced from news content and adjust to 1500 input elements (100 dimensions x 15 words from headline). These input feed to the neural network usually consists of neurons called hidden layers which could have several neurons connected to the input layer. The relationship between each connection is normally in the form y = *W**x+b where *W* is the weight and *b* is bias. Fully connected means that all 1500 input nodes connect to every neuron in the hidden layer which could be any arbitrary number of neurons. The output of the first hidden layers can then be fed to another hidden layers until the last layer which has the same amount of output vectors which are the ten most liquid stocks listed in the Stock Exchange of Thailand (SET). These stocks are ADVANC, BANPU, BEM, CBG, CPALL, CPF, IVL, KBANK, PTT, and TOP, whose price data are from BizNews. The data is fed forward through the neurons via the input layer and calculate to get the predicted output which is then compared to the real output. The difference is called loss function or cost function and will be calculated to provide adjustments to all the weights and biases in the network to minimize the loss. This process is called backpropagation.

To improve the accuracy and performance of neural network, many techniques have been developed to the simple fully connected architecture. The most popular techniques include convolutional neural network [LeCun et al (2008)] and recurrent neural network [Sak et al (2014)].

Convolutional Neural Network is inspired by the mechanism of animal visual cortex where the image is not processed as a whole but the brain has several groups of neurons to process different filters grouped by certain aspects such as edge detection, contrast, outstanding hue, etc. In machine learning, the biological process can be archived by using *n*-dimensional matrix called filters (or kernels) applied to each segment of the input data (convolute) and recreate a new data which contains certain aspect of the original data.

The data is then downsampled using pooling technique to look at the convoluted data and pick out a data that represent that block. This helps reducing the amount of computational tasks and avoid overfitting as the result is more random due to the nature of the pooling. Pooling can select the maximum data in the block (maxpool) or average of the block (avgpool).

Another simple technique is to just drop some of the data, this simple process of ignoring output inside the hidden layers helps generalizing the network and improves overfitting issue [Srivastava (2014)].

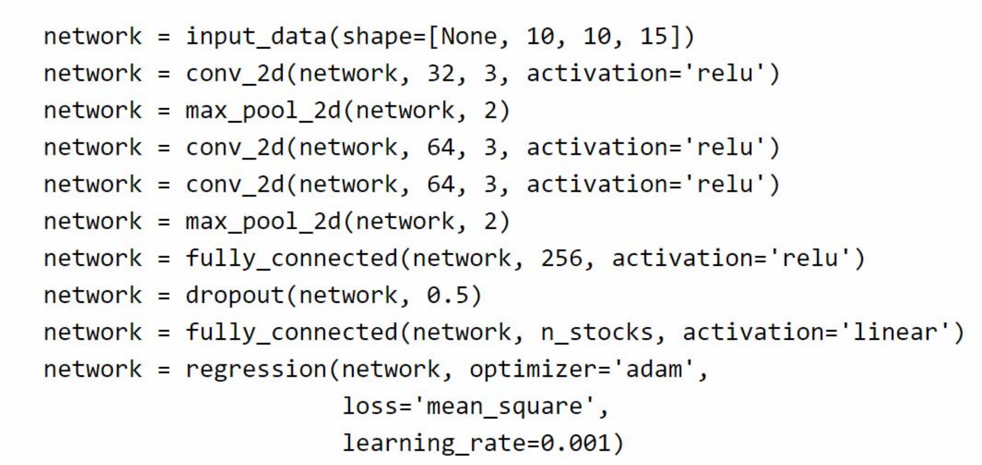
Once the output nodes are calculated, the result can then be applied with an activation function to convert a neuron's weighted input to its output. The activation function of each neuron usually yields one output where rectifier functions such as *tanh, sigmoid, softmax, softplus*, and *relu* are suitable to enhance the output for each neuron.

To teach the network, the machine has to compare the predicted output against the real output. The function to calculate the difference is called loss function which is used to calculate the objective. In our case, the mean squared method provides the most favorable training due to the linear nature of the outputs.

The weights and biases parameters inside the network can then be adjusted using optimizer algorithms. The most popular ones include *Stochastic Gradient Descent* (SGD), *RMSProp, Adam, Momentum, AdaGrad, Ftrl, AdaDelta*.

I arbitrarily define eight layers in this study with details described in python code shown in Figure 2. Our network has input 1500 nodes which have been reshaped into image-like tensor of 10 x 10 x 15. It then sends all inputs to the first convolution layer which contains 32 filters. Each with the size of 3. The result is fed to a *max_pool* layer, another 2 convolution layers with 64 filters each, then connect to the fully connected layer before dropping 50% of the outputs and feed via another fully connected layer to create the output result.

The learning process uses mean squared method with *Adam* optimizer. The network parameters will be adjusted by the parameter learning rate. If the learning rate is too low, it could take a long time to find the optimal parameters whereas if the rate is too high, the machine might skip over the optimal parameters and never find the right ones.



```
network = input_data(shape=[None, 10, 10, 15])
network = conv_2d(network, 32, 3, activation='relu')
network = max_pool_2d(network, 2)
network = conv_2d(network, 64, 3, activation='relu')
network = conv_2d(network, 64, 3, activation='relu')
network = max_pool_2d(network, 2)
network = fully_connected(network, 256, activation='relu')
network = dropout(network, 0.5)
network = fully_connected(network, n_stocks, activation='linear')
network = regression(network, optimizer='adam',
                     loss='mean_square',
                     learning_rate=0.001)
```

Figure 2 Python Code to be Run by TensorFlow

There are 10 nodes in the output layer representing the ten most liquid stocks in SET50 index. The training runs from January 1, 2016 to June 30, 2016 and the out-of-sample test run from July 1 - 31, 2016. The stock market movement during such period can be shown in Figure 3. It is apparent that the market is bullish in the test month. Stock trading during July 2016 yielded a daily return of 3.75 percent on an average.

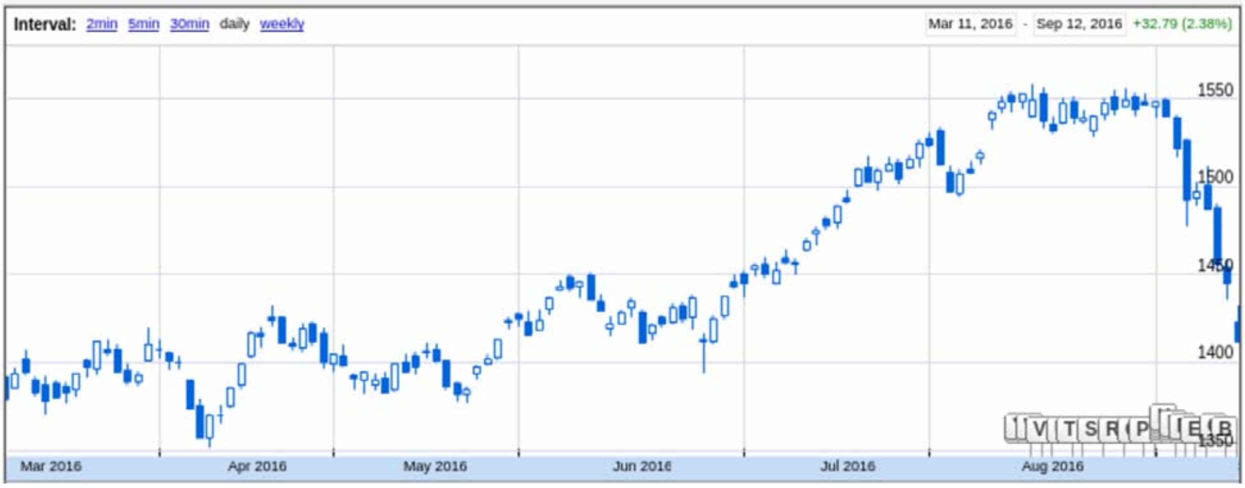


Figure 3 SET Index

IV. Results

Based on 78,995 news from NewsCenter, this study finds that news are usually announced in three periods a day. Figure 4 shows that news are usually released during off-trading hour judging from the fact that the trading hours of the stock Exchange of Thailand are 10.00-12.30 for the morning session and 14.30-16.30 for the afternoon session. This is understandable that the companies intend to lessen the impact on their stock prices' volatilities.

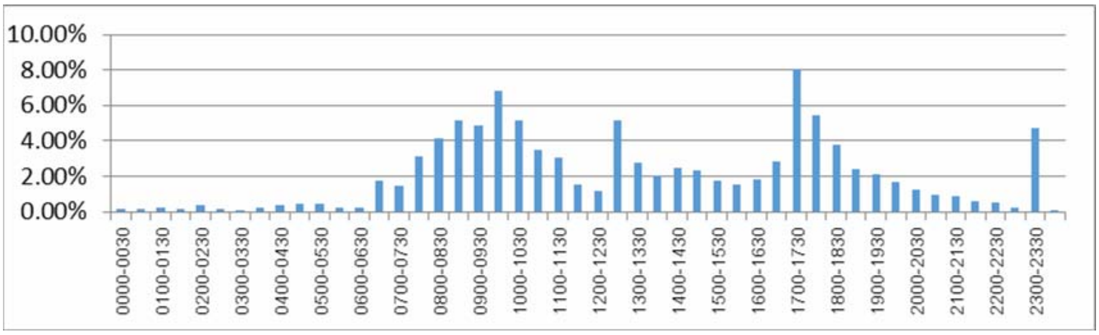


Figure 4 News Release Distribution

To test the predictability of the model, trading strategies with different holding periods are used. First, one-day holding period is tested. This means that after the news was announced, I buy (short) the stocks that the model predicts that the price will rise (fall) at the end of the day. The threshold is set that change in price should be greater than certain value to activate trading. The result in Table 1 defines the threshold as 1 percent for 1-day trading, and 0.1 percent for all short interval trading strategies.

Table 1 Results of Trading *n*-minute after News Release Suggested by the Model

	Long 5-minute (1)	Short 5-minute (2)	Long 10-minute (3)	Short 10-minute (4)	Long 30-minute (5)	Short 30-minute (6)	Long end of the day (7)	Short end of the day (8)
N	2000	2000	2000	2000	2000	2000	2000	2000
mean	0.001307	0.002085	-6.5E-05	-0.00451	0.00447	-0.00226	0.101732	0
Std.	0.004806	0.005088	0.005453	0.006562	0.013535	0.009127	0.041705	0
Min	-0.02573	-0.01974	-0.03442	-0.04554	-0.04608	-0.04766	0.029552	0
25th Per	0	0	0	-0.00769	-0.00349	-0.00606	0.075213	0
50th Per	0	0	0	-0.00769	0.004313	-0.00116	0.087408	0
75th Per	0	0.006061	0	0	0.005894	-0.00116	0.155668	0
Max	0.017259	0.042938	0.02833	0.038816	0.076557	0.060658	0.174302	0

Column 7 and 8 in Table 1 show that the model only suggests buy strategy that yields 10% daily return on the average and it is significant. No daily shortselling is suggested. This means the model can satisfactorily foresee increasing trend of the market in July 2016. When the holding period is shorten to 5 minutes, 10 minutes, and 30 minutes, the result suggests only positive returns of 5 minutes trading strategy and negative return from short return of 30 minutes trading strategy are significant. Since 30 minutes strategy yields negative but significant return, I ignore it from the analysis. Assuming that there are four and half trading hours in the Stock Exchange of Thailand for one day, this translates 5-minute strategy returns in column 1 and 2 into 7 percent and 11.2 percent on a daily basis for long and short strategies, respectively, or a combined return of 18.2 percent if trader deploys both strategies.

V. Conclusions

This study is the first that uses word vector to translate Thai news into a quantifiable representative vector. It is the first that applies deep learning to turn word vectors into predicting stock return in a short interval. The results suggest the possibility to exploit market adjustment to gain abnormal return. It shall be noted that although 5-minute trading strategy outperform daily trading, most returns come from “short” strategy. Considering that the stock price movement in July 2016 is obviously uptrend with the daily average return of

3.75%, the deep learning approach can ride on the volatility and enable investors to gain abnormal profit from the market correction.

The main objective of this study is to answer if the public news can predict stock performance and can the deep neural network analyse and detect the news content. Although the results are positive, it could be improved in a number of ways before the machine could be put into the actual trade. More hidden layers could be added to the network to improve the accuracy and nature of the prediction. The system needed to be able to analyze all major stocks in the market and identify the correct ones. Another interesting concept is to ensemble news, fundamental, and technical analyses to predict the market movement.

References

Chattupan, A., and Netisopakul, P. (2015). "Thai Stock News Sentiment Classification using Wordpair Features." *29th Pacific Asia Conference on Language, Information and Computation*, 188 - 195.

Ding, X., Zhang, Y., Liu, T., and Duan, J. (2015). "Deep Learning for Event-Driven Stock Prediction." *the 24th International Conference on Artificial Intelligence*, 2327 - 2333.

Enke, D., and Thawornwong, S. (2005). "The Use of Data Mining and Neural Networks for Forcasting Stock Market Returns." *Expert Systems with Applications 29 (4)*, 927 - 940.

Fai Wong, F., Liu, Z., and Chiang, M. (2015). "Stock Market Prediction from WSJ: Text Mining via Sparse Matrix Factorization." *Enterprise Applications and Services in the Finance Industry,* 13 - 28.

Feldman, R., Rosenfeld, B., Bar-Haim, R., and Fresko, M. 2011. "The Stock Sonar — Sentiment Analysis of Stocks Based on a Hybrid Approach." *Proceedings of 23rd IAAI Conference on Artificial Intelligence (IAAI-2011).*

Fama, E., Fisher, L., Jensen, M., and Roll, R. (1969). “The Adjustment of Stock Prices to New Information”, *International Economic Review, 10,* 1-21.

Fama, E. (1970). “Efficient Capital Markets: A Review of Theory and Empirical Work”, *Journal of Finance, 25,* 383-417.

Google Brain Team (2016). "TensorFlow: A System for Large-scale Learning".

Heston, S., and Sinha, N. (2016). "News versus Sentiment: Predicting Stock Returns from News Stories." *FEDS Working Paper No. 2016-048.*

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). "Gradient-based Learning Applied to Document Recognition." *Proceedings of the IEEE. 86 (11),* 2278–2324.

Luss, R., and d'Aspremont, A. (2008). "Predicting Abnormal Return From News using Text Classification." *Quantitative Finance 15 (6),* 35 - 44.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ICLR Workshop.*

Nguyen, T., Shiraia, K., and Velcinb, J. (2015). "Sentiment Analysis on Social Media for Stock Movement Prediction." *Expert Systems with Applications 42 (24),* 9603 – 9611.

Sak, H., Senior, A., and Beaufays, F. (2014). “Long Short-term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling.” *Proceedings of Interspeech*, pp338-342.

Schumake, R., and Chen, H. (2009). "Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System." *ACM Transactions on Information Systems (TOIS) 27 (2)*, 1 - 29.

Srivastava, N., Hinton, C., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of* Machine Learning Research. 15 (1), 1929–1958.