Stock Market Investment Using CNN Model

Abstract

This was originally an academic project advised by Mr. Yu-Shiang Lin, a PhD candidate at the Graduate Institute of Electrical Engineering, National Taiwan University, and after several discussions and the inclusion of some experts and consultants in Finance, the project turned into an entrepreneurial venture. The project aims at helping people make better investment decisions in stock market by referencing to objective metrics, such as the change of technical indicators over a certain period of time, instead of irrational impulses or gut feelings.

The project is currently incubated by and in collaboration with Guosen Securities's FinTech team, and this statement covers the work of our research and the results on investment simulation.

Introduction

This project applies Convoluted Neural Network (CNN) structure and methods presented in an article published in *Applied Soft Computing* entitled "Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach", and since not all information is presented with details in the article and it requires better financial results than that of the work in the article to create a viable business model, we conducted plenty of experiments to research on several topics to suit our own purposes. The research experiments mainly focused on normalization methods, solutions to data imbalance, and pattern discerning on stock data.

The rest of the statement is structured into seven sections – Overall Work Briefing, Research on Normalization, Research on Data Amount, Research on Data Imbalance, Patterns and Findings, Conclusion, and Future Work.

Overall Work Briefing

Instead of 15 stock trading technical indicators presented in the article, we used only 11 indicators, suggested by our financial consultants and in order to speed up the overall calculation runtime. 11 frequently used stock technical indicators (e.g. RSI, MACD, EMA, etc.) with their value within 11 different timeframe to create a 11x11 square matrix for every trading day. Each 11x11 square matrix is viewed as an image to be sent to the CNN model for pattern recognition and classification. There are 3 classes: Sell, Buy, and Hold. Classification for training data is done by taking reference to past data of 5 days and future data of 5 days (i.e. If classifying data on 1 of January, 2002, then data on days from 2 to 6 of January, 2002 will be taken). If the day to be classified has the highest closing price within this 11-day timeframe, it'll be marked as 'Sell'; If the opposite, it'll be marked as 'Buy'; And any other value in between the highest and the lowest will be classified as 'Hold'. Then, the CNN model will take the 15x15 matrix, as an image, into calculation and eventually predict the transaction action (i.e. 'Buy', 'Sell', or 'Hold') of the day.

Two testing methods are used to examine the accuracy of the results. The first one is of mathematics and statistics, where Recall, Precision, and F1 Score are calculated and analyzed. The second one is of investment simulation, where predicted transaction actions are used to conduct transactions with real data in stock market and the Return of Investment over 12 years are recorded.

In investment simulation, we take the same method mentioned in the article. When performing a transaction action of a day, 100% of the asset would be transacted (i.e. When buying, all the capital will be used to buy stocks; When selling, all stocks will be sold to cash in; When holding, no transaction would be performed). In addition, if there are multiple same transaction actions consecutively, only the first transaction action would be performed until the next different transaction action. For example, if the prediction on transaction action is 'Buy' on Jan. 1, Jan. 2, and Jan. 3, then only the 'Buy' action will be performed on Jan. 1. And the results of the investment simulation were compared, across 12 years, to that of the *Buy and Hold Strategy* (Buying at the beginning of the year and selling at the end of the year without any other actions throughout the year) and that of the optimal situation (Always buying at the lowest price and selling at the highest price).

In the article, 6 years were taken into a group for training and testing. If year 2009 is to be tested, then stock data from 2004 to 2008 will be used as training data and stock data in 2009 will be used for testing, and the result of investment simulation and mathematic calculation will be recorded down. In our project, we tried both a 6-year method, just like the way conducted in the article, and an all-available-data method (all available data before the testing year is taken into training), over a 13-year period.

In this statement, the stock used for demonstration is JPM since it is of large scale that is suitable for quantitative analysis, it has been fluctuating over the past years (so that it provides better learning materials for the model), and there is detailed results shown in the article (so that comparison can be conducted).

Research on Normalization

Data normalization is a crucial part of performing machine/deep learning since different types of data may be of different scale, and sometimes a too large or too small value, due to the nature of the data itself, might affect the overall prediction and calculation. However, compressing numbers into a unified scale might, on the other hand, compromise the actual representation of the data.

Our current finding is that **the results would be slightly better without implementing any normalization methods**. Perhaps, this is because compressing technical indicators might really compromise the actual representation of the data, especially for those indicators that have a wide range. Also, it is possible that normalization in our case would not be necessary since almost all indicators have similar range (usually ranged from 0 to 100 or -100 to 100).

Research on Data Amount

Machine/deep learning requires a great deal of data to learn well and make reliable predictions, hence, the amount of data available and used is quite an important issue. We tried different amount of data on training as mentioned in the overall briefing, 6-year methods (taking data only from previous 5 years, as mentioned in the article) and all-available-data methods (taking all available data before the testing year).

Our current finding is that **training and testing with only data from previous 5 years generates better results on most of the stocks**, and the occurrence of this pattern is, with the experience of our financial experts, highly because of the fact that stock trading actions and results from too long ago would not have too much impact on recent stock trading activities, especially when many other deciding factors have changed greatly overtime, such as economics, management team and strategy of the companies, legislations, policies, etc.

Research on Data Imbalance

It is normal for stock investors to only trade a few times throughout a month or even a year because of the nature of the stock market. Highest or lowest prices don't occur on a frequent basis and it also costs quite a lot to trade too frequently due to the transaction fee. Therefore, in our data labelling, around 90% of the data would be labelled ‘Hold’ and only around 10% of the data would be labelled ‘Buy’ and ‘Sell’, with around 5% each, and this caused a serious data imbalance problem.

We tried mainly 3 methods to tackle such issue – Scale Up (Adding noise to existing data to creating more similar data of ‘Buy’ and ‘Sell’), Scale Down (Randomly reducing data of ‘Hold’ to reach a balance between the 3 categories), and Weight Adjusting (Adjusting the weight applied in calculating loss function by giving ‘Buy’ and ‘Sell’ more weight to offset their scarcity in the data pool).

Our current finding is that, in our cases, **the Weight Adjusting method has been rendering stable and generally better results**. The Scale Up method is prone to causing overfitting, with higher training performance, but no commensurate testing results. The Scale Down method causes the total amount of data would be too small, which is not enough for a proper deep learning training. The Weight Adjusting method, however, serves as a middle ground of the two other methods, as no additional overly similar data will be added and no data will be removed, and our experiment further found out that setting the weight of data entries labelled as ‘Buy’ and ‘Sell’ to around 40 to 50 would give a significantly greater result.

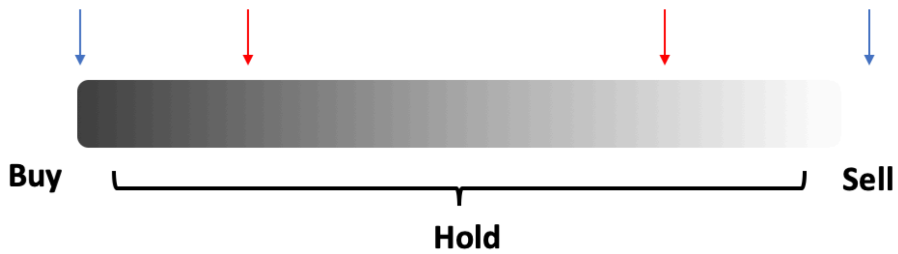
Patterns and Findings

The most important finding in our research is that deep learning models still cannot categorize ‘Buy’, ‘Sell’, and ‘Hold’ with high precision due to the nature of stock price and the way our training data is processed. And this is because of 2 factors as follows:

1. Making decision on stock trading actions is a task without a standardized rule. Unlike categorizing pictures of dogs and cats (there is a set of very standard, objective, and clear criteria on the definition of the idea of dogs and cats, such as their physical attributes), there is really no set of standard and clear criteria on when to buy and sell

stocks exactly for the model to learn and for human to later reinforce the learning process (everyone knows and wants to buy at the lowest price and sell at the highest price, but nobody or no metric can tell exactly when the highest or lowest price would come, let alone stock market is supported by supply-demand mechanism with countless of factors that cannot be subsumed in deep learning model training).

2. The model seems to notice the pattern before the highest price and the lowest price, i.e. there is an increase before arriving at the highest price and a decrease before arriving at the lowest price; Yet, the model seem not to be able to spot the highest or the lowest price point. This is understandable because only when a trend of decrease starts can we know that the pinnacle has emerged and there's just no way to know if any current point is the highest point (and vice versa for the lowest price point). At the same time, when providing data for the model to predict, only current or past data would be available, hence, unless the model can learn a general and stable enough pattern from current and past data (which is the ultimate goal of the project, but now seems to be the very obstacle to precise prediction), the model can only make guess on the highest or the lowest price point when it discern an increasing or decreasing pattern.



The image above provides a visualization of the issue. When we label a piece of data 'Buy' or 'Sell', we are only providing the model with the a relatively extreme case (the 2 extreme points marked by blue arrows), rather than a set of standard and clear rules, and hoping that the model would learn to spot such extreme cases. In addition, when the model is provided a piece of data that doesn't represent the highest price or the lowest price (other points in between the 2 extreme points marked by red arrows), the model would have a hard time knowing if such point is an extreme point or just other points in between, since we didn't give specific boundaries or rules about those extreme points when training (and we, as human beings, actually can't do so simply because there's no standard or clear rules of the idea of when to buy or sell stocks) and the model can only get access to current and past data (and we human can only provide such data). Hence, there will be several consecutive prediction of 'Buy's and 'Sell's over a short period of time with an increase or decrease trend in price, while those predicted as 'Buy' and 'Hold' should actually be categorized as 'Hold'. (Check images below, where 1 represents 'Sell', 2 represents 'Buy', and 0 represents 'Hold')

date	price	predicetd label	original label
2018-04-27	109.400002	2	0
2018-04-30	108.779999	2	0
2018-05-01	108.779999	2	0
2018-05-02	107.919998	2	0
2018-05-03	107.239998	2	2

Consecutive prediction of 'Buy' during a trend of decrease in price

date	price	predicetd label	original label
2018-08-03	117.089996	1	0
2018-08-06	117.120003	1	0
2018-08-07	117.550003	1	0
2018-08-08	117.790001	1	1

Consecutive prediction of 'Sell' during a trend of increase in price

<u>Conclusion</u>

Currently our model has been performing better in investment simulation, about twice the investment return, than the model presented in the article (Image 1 & 2) and better than average human investors and the *Buy and Hold* Strategy (Image 3) with a **6-year method in terms of amount of data for training**, **the Adjust Weight method for dealing with data imbalance**, and **without any normalization.**

However, in terms of mathematic and statistical performance, the results of ours are still not as good as those in the article (Image 3 & 4). In addition, we hope to achieve a 30-time investment return over a 13-year period. Hence, more work and further research, especially in cost reduction in overall runtime, fine tuning on training parameters, deployment of CNN layers, or even potential combination of different deep learning models, will be needed in order to arrive at such a demanding goal.

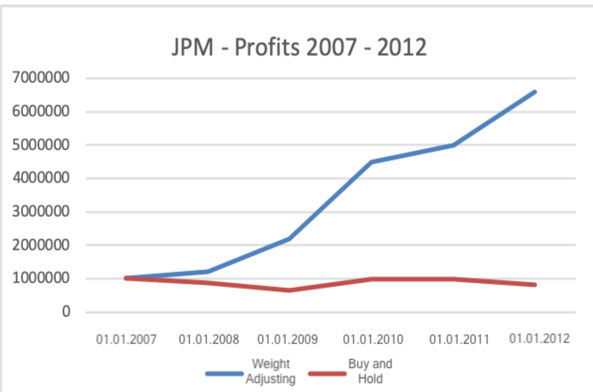


Image 1 – Results of Weight Adjusting have almost 7 times of Return of Investment from 2007 to 2012

Image 2 – Results shown in the article have almost 4 times of Return of Investment from 2007 to 2012

year	optimal_start	optimal_final	optimal_return	pred_start	pred_final	pred_return	bah_start	bah_final	bah_return
2007	1000000	1706544.251	71%	1000000	1199266.5	20%	1000000	876674.36	-12%
2008	1706544.25	4141455.967	143%	1199266.53	2197719.9	83%	876674.4	651461.54	-26%
2009	4141455.97	19775872.31	378%	2197719.9	4498116.5	105%	651461.5	978366.57	50%
2010	19775872.3	31813952.5	61%	4498116.49	4994582	11%	978366.6	993997.93	2%
2011	31813952.5	34971242.53	10%	4994582.01	6596899.9	32%	993997.9	808600.32	-19%
2012	34971242.5	46066438.55	32%	6596899.87	7959413.9	21%	808600.3	1081550.5	34%
2013	46066438.5	68349044.25	48%	7959413.9	9586970.8	20%	1081551	1427560.2	32%
2014	68349044.2	138260224.9	102%	9586970.79	10848241	13%	1427560	1484564.6	4%
2015	138260225	233539443.8	69%	10848241.5	11393145	5%	1484565	1556200.9	5%
2016	233539444	352340574.8	51%	11393145.4	13661946	20%	1556201	2286555.8	47%
2017	352340575	616685296.7	75%	13661946.1	13967421	2%	2286556	2962511.5	30%
2018	616685297	972978382.2	58%	13967421.4	14375324	3%	2962511	2812275	-5%
2019	972978382	2231375088	129%	14375323.9	16855463	17%	2812275	3963290.4	41%

Image 3 – Results of Weight Adjusting (pred in the chart) are about 4 time more profitable than Buy and Hold (bah in the chart), while optimal results have a 2000+ times return (optimal in the chart)

Total Accuracy: 0.58			
	Hold	**Buy**	**Sell**
Recall	0.55	0.80	0.81
Precision	0.95	0.22	0.18
F1 Score	0.70	0.34	0.29

Image 3 – Mathematic and Statistical Results on Overall Stock Prediction in the Article

Total Accuracy: 0.49			
	Hold	**Buy**	**Sell**
Recall	0.46	0.86	0.79
Precision	0.96	0.16	0.13
F1 Score	0.61	0.26	0.21

Image 4 – Mathematic and Statistical Results on Overall Stock Prediction of Our Team

Future Work

There are currently two main paths for future endeavor:

1. Try applying more mathematic and advanced deep learning techniques to improve our model's feature learning.
2. Try labelling data with attributes of a fixed or a more objective rule, such as 'Closing Price Will Increase on the Next Day' and 'Closing Price Will Decrease on the Next Day'. The model would only produce objective predictions (predicting 'Increasing' or 'Decreasing' is objective, while predicting 'Buy', 'Sell', or 'Hold' is subjective), and the actual transaction action can be left for investors to decide.