COMP576 Final Project Report

Image-Based Foreign Exchange (FX) Movement Prediction

Wei-Chong Chen

Introduction

In financial market, future price prediction is always a hot topic to explore as shown in Figure 1. However, future price prediction is a trivial and challenging work with small use.

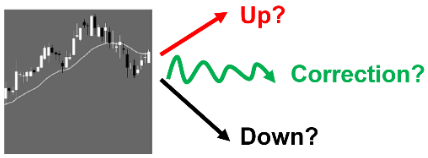

Figure 1

In the project, I would like to analyze and predict the future trend of Foreign Exchange (FX) market through deep neural network. Instead of predicting the exact price of the item, I want to predict the trend (bull, bear, and correction) based on the indicators, structure and pattern of the item so that the investors can have a clear mind on whether to long, short or even hold the item based on the result. Therefore, the goal of the project is to give the investors a better idea on how to deal with the item of interest like Figure 2.

Figure 2

Background

We all know that market price is extremely hard to predict since there are so many factors influence the fluctuation, and there are also some interior reason behind a certain move. Some people even believe that the market behavior is purely a random walk [9].

People who believe that the market has its own pattern to follow and further predict facilitate the idea of machine learning and deep learning [7]. There are many papers discussing about market predicting with deep neural network [1], [3], [4], and [5]. However, the authors mainly focus on predicting the exact number of the market which gives no clue on what the future trend of the market, furthermore, the results could be overfitting at some point.

I am a big fan of indicators, structure and pattern of the market. In them, I believe there are clues of the behavior of the market. Therefore, I would like to analyze them with deep neural network, and manage to predict the future movement of the market. There are some papers backing up the idea of indicators, structure and pattern [2], [6], and [8]. Inspired by the papers and my belief, I will implement the prediction based on those features to predict market movement.

Methods

Moving Average (MA):

In market analysis, MA plays an important role since it is a straightforward indicator which unveil market's main direction in certain period. Besides, MA is relatively easy to interpret compare to the other indicators. Therefore, investors would like to determine the movement of the market with the help of MA.

$$MA_{i+1} = \frac{1}{period} \sum_{k=i-n+1}^{i} MA_k$$

However, MA would be failed if the market goes to correction phase because when correction happens, the direction of the market is equivocal, and usually, the price tends to consolidate near MA indicator, resulting in several fake-out breakout of MA (Figure 2 Correction). To overcome this, people tends to use larger period of MA which will wipe out some noise of the market, however, not all the fake-out breakout can be ignored, what's more, MA will become slow and cannot reflect the direction of current market.

Thus, how to determine the breakout is not fake is an interesting problem. And this problem could be solved with consideration of the market structure or pattern. Some inherent structures/patterns are shown effective sometime, for example, the completion of head and shoulders pattern indicates big up/down movement of the market (Figure 3).

Figure 3

Another example, according to Elliot's wave theory, the wave structure is how the market constructs (Figure 4). And there are more theories manage to describe the market structure/pattern, trying to analyze the market movement.

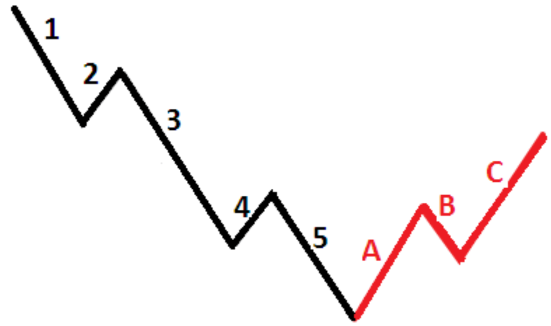

Figure 4

Based on these, I would like to see if there are some structures/patterns that imply a big up/down movement. And meanwhile, exclude the fake-out breakout of correction. Namely, in combination with MA and structures/patterns, we might be able to determine the movement of market more effectively.

Because I want to examine the market from the image point of view, in this project, I manually capture 1199 figures from website (https://www.tradingview.com) which is composed of 385 up-trend, 385 down-trend, and 429 correction figures. It is time consuming yet interesting process. And the market movement determination are as follows.

From the candlestick chart, because I have a eagle view, I know what movement it would be in the future. When the candlestick is breaking out MA and the close market price is above MA as well, I will claim it as up movement as long as the future movement has a big upward movement. And similarly, I will capture down movement figure when the candlestick is breaking out MA and the close market price is below MA. For correction, when the market is up movement, during the movement, is the downward breaking out does not last long and far enough, I will identify it as correction. If the correction direction is same as the main direction, I will not collect it. For example, if it is up movement market, during the entire up movement until the end, all the upward breakout will be ignored. The capturing flow is shown in Figure 5.

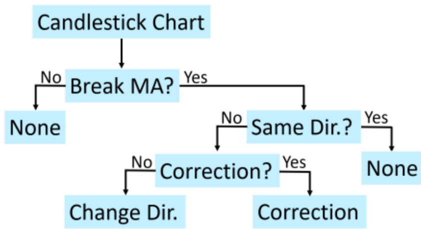

Figure 5

This capture process mainly tries to identify the early stage of up/down movement while including correction into consideration.

With captured pictures, we can further train the neural network. For the first idea, I use convolutional neural network (CNN) to predict since it has better stability and better performance on images in general, and then I will try to compare with recurrent neural network (RNN), and its extension: long short-term memory (LSTM), and gated recurrent units (GRU) to see if the factor of time or feedbacks will induce better result.

CNN:

CNN is a feedforward neural network, whose artificial neurons can respond to neighbor units in certain part of areas. CNN consists of one or more convolutional layers, an upper fully connected layer and pooling layers. Compared to other deep learning structures, CNN can do better when processing image, whereas CNN needs less parameters to take into consideration. Also, CNN can be used in back propagation practice.

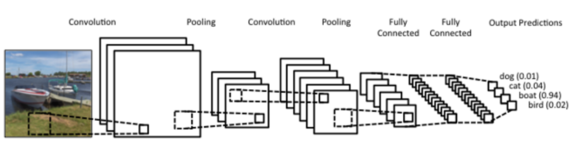

Figure 6

RNN:

RNN is a network where connections between nodes form a directed graph along a sequence. This property allows it to exhibit temporal dynamic behavior for a time sequence. Unlike feedforward neural networks such as CNN, RNN can use internal state/memory to process sequences of inputs. Therefore, RNN is suitable to tasks such as unsegmented, connected handwriting recognition.

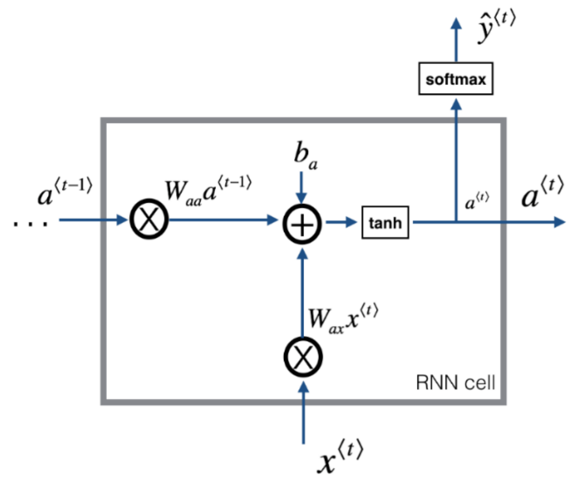

Figure 7

LSTM:

LSTM (Figure 8), an extension of RNN, can be used as a building component or block for a bigger RNN. It contains a cell connected through layers responsible for important values to remember over time.

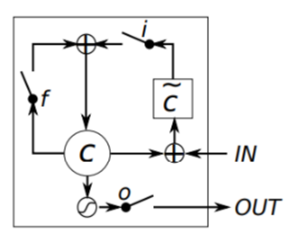

Figure 8

GRU:

GRU (Figure 9) is another extension of RNN. Similar to LSTM, GRU has gated units to modulate the flow of information inside the unit. The main difference between GRU and LSTM is that GRU doesn't have separate memory cells, resulting in higher speed. Nevertheless, the performance of GRU might not as good as LSTM since it has less information than LSTM.

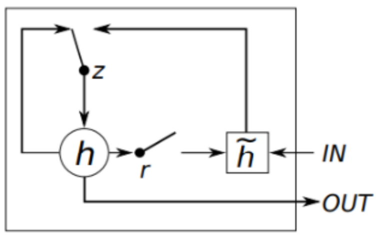

Figure 9

The feeding inputs are the raw figure captured on 'TradingView' with predifined condition. The output should be either uptrend, downtrend, or correction of the future market movement. Ideally, the result should give advice to the investors on whether they should long, short or even hold the item.

Numerical Results

In the simulation, the total dataset is 1199 captured figures with shuffled 1079 figures as training dataset and 120 figures as testing dataset. The key idea of the simulation it to identify the 'Gold' market future movement with candlestick and MA, plus the idea of structures/patterns.

The batch size is set as 128, and apparently, the number of classes to classify is 3. The optimizer is chosen as gradient descent with learning rate 0.001. However, due to memory constraints, the real size (389 by 389) of captured figure cannot be successfully trained on free training platform (CoLab+GPU) provided by Google. The approach to overcome this is to sacrifice resolution by resizing figures by 40%.

In CNN, the epoch is 100, and the order of layers are as follows: Conv2D: (32, 3*3, relu)➔ Conv2D: (64, 3*3, relu)➔ Max pooling: (2*2)➔ Dropout = 0.25➔ Dense: (128, relu)➔ Dropout = 0.5➔ Dense: (3, softmax)

In RNN-based, the training step is 10,000 steps, I adopt BasicRNNCell for RNN, and LSTMCell (forget bias = 1.0) for LSTM, GRUCell for GRU, which are the packages provided under tensorflow.nn.rnn_cell. For simplicity, all the other parameters are set as default.

One thing to mention, all the captured figures are captured chronologically, therefore, most of the figures has some overlapping portion related to previous figures. That's why I will go for RNN-based system for a possibly better result.

To have a better look, I put CNN and RNN-based together, and map 100th epoch of CNN to 10,000th step of RNN.

Figure 10 reflects the training and testing loss among all networks. We can see that CNN has smallest training loss while have largest testing loss which implies overfitting of CNN. And for RNN-based network, although they converge, they cannot reduce training and testing loss any further. The reason behind could be that the figures are tough to train. In Figure 11, again, CNN has perfect training accuracy, however, performs worse than the others in testing dataset. What's more, we can find that no one approaches prediction accuracy higher than 70%.

My insight of this is: 1) number of dataset (captured figures) is not enough 2) the structure/pattern of correction is extremely challenging to determine. Hence assigning the upward correction as up-trend and downward correction as down-trend has lower risk/loss compared to claim it as correction.

During poster session, luckily, Dr. Ankit is one of my graders. He showed interests toward my project and gave me some advices: 1) do regression on price, 2) try only 2 classes (up, down), 3) try different parameters to lower training loss for RNN-based system.

Because my project based on image input instead of actual price number, I save regression for future work.

Hence I first try 2 classes, while managing to decrease training loss at the same time. I use 256 neurons (512 neurons does worse) opposed to 128 neurons previously. In addition, I train them more times to 20,000 steps. Interestingly, I find that training loss drops to zero in all case in Figure 12. However, basic RNN seems to be overfitting. For accuracy, all the training accuracies approach to 1. In Figure 13, we can find that LSTM and GRU beat CNN with high testing accuracy. And not surprising, basic RNN dose worse with only 0.6 accuracy.

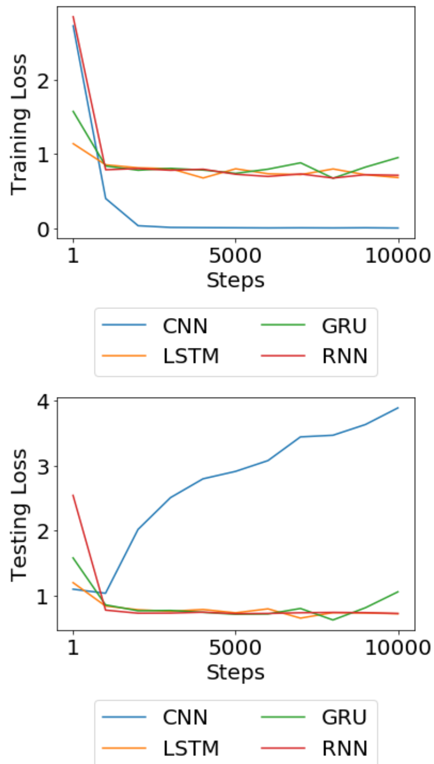

Figure 10

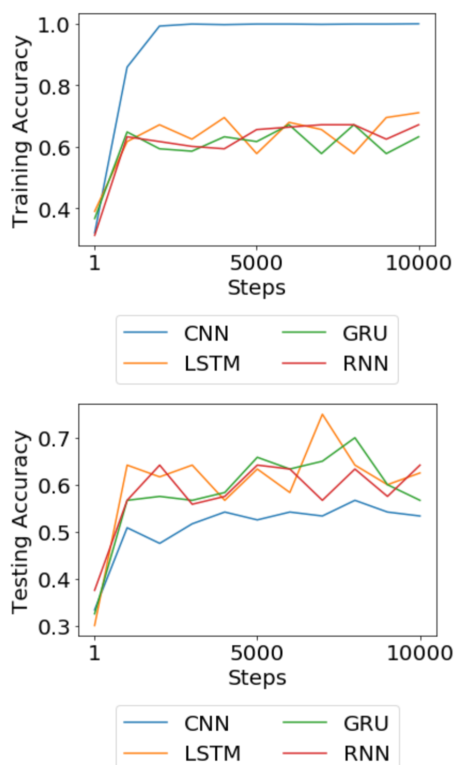

Figure 11

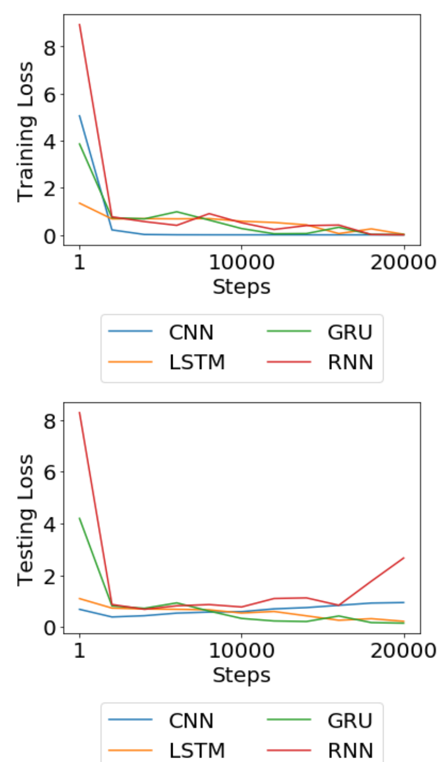

Figure 12

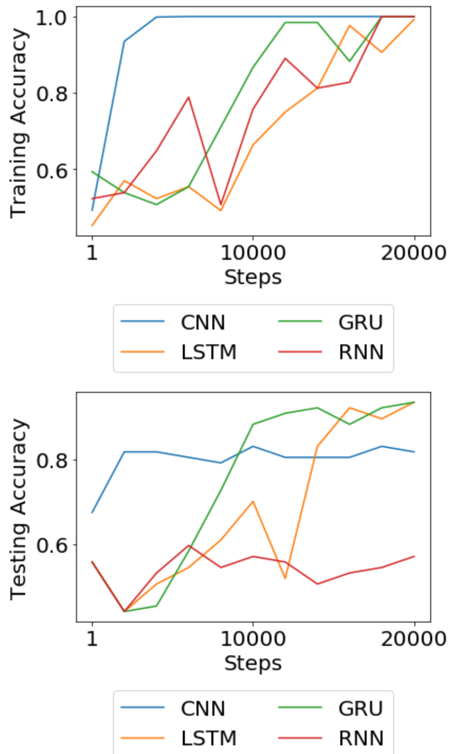

Figure 13

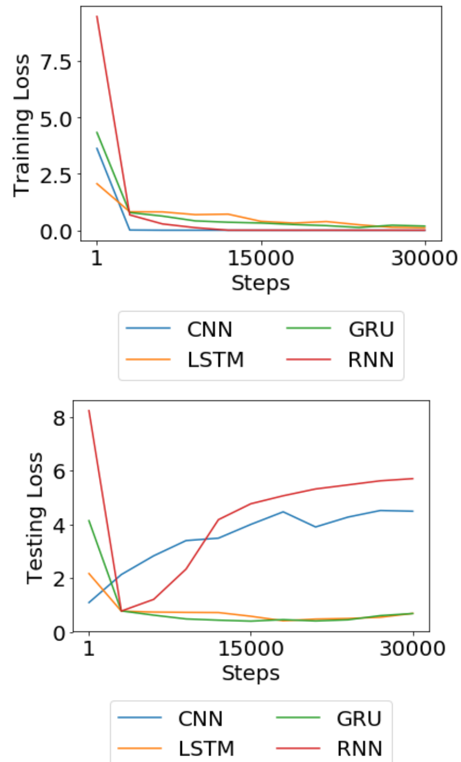

Figure 14

With better parameter, I train the network for 3 classes results (up, down, correction) with 256 neurons with 30,000 steps. We can find that the training loss in Figure 14 goes to zero for all cases. Again, CNN and basic RNN faced with overfitting. In Figure 15, The training accuracy for all systems are performing well. However, basic RNN and CNN cannot do better than 0.6, while basic RNN is around only 0.5. Although 0.5 is still not bad for 3 classes, LSTM and GRU can approach to 0.8, which is a promising system to me!

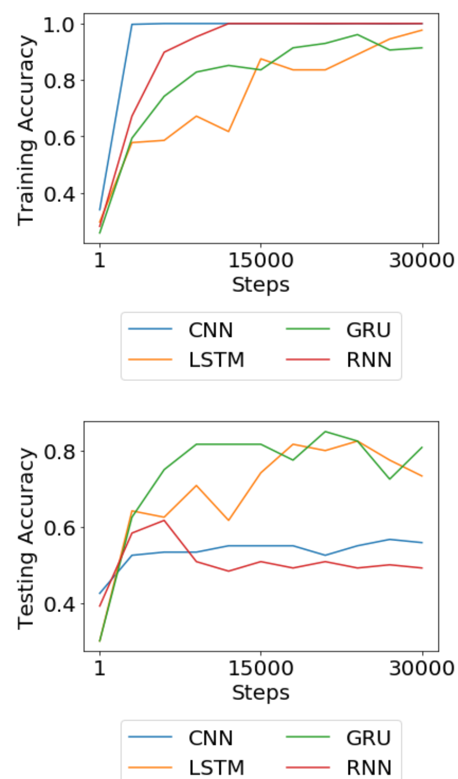

Figure 15

Conclusions & Future Works

Market movement prediction is still tough and challenging. Although I get 60% prediction accuracy on market movement with RNN, it cannot do better with even 10,000 steps. However, with 256 neurons and more training steps (30,000), the performance of LSTM and GRU can approach 0.8 thanks to the overlapping property between neighbor figures. However, CNN cannot do well in this case although it has high reputation on recognize images. And basic RNN cannot do well since it remembers everything, resulting in overfitting.

To become better, we could try with larger number of training dataset to define correction movement. In addition, performance might be better by introducing other indicators.

The reason could be that the very important feature only lies in the right end of the figure (where candlestick breaks MA), from observation, CNN and RNN tend to classify correction to either up-trend or down-trend more other than correction itself. And resizing could be a key factor of missing the correction feature.

The inherent structures/patterns seem to be tough to identify, I think if we can pre-define some type of structures/patterns first could be a big help.

References:

[1] Hedayati Moghaddam, A., Hedayati Moghaddam, M., Esfandyari,M., 2016. Stock market index prediction using artificial neural network Predicción del índice del mercado bursátil utilizandouna red neuronal artificial. J. Econ. *Finance Adm. Sci*. 21(December (14)), 89–93.

[2] O.B. Sezer, M. Ozbayoglu, E. Dogdu, A deep neural-network based stock trading system based on evolutionary optimized technical analysis parameters*, Proc. Comput. Sci*. 114 (2017) 473-480.

[3] T. Fisher and C. Krauss, Deep learning with Long Short-Term Memory networks for financial market predictions, *FAU Discussion papers in Economics*, (2017).

[4] Hiransha, M.; Gopalakrishnan, E.A.; Menon, V.K.; Soman, K.P. NSE stock market prediction using deep-learning models. *Procedia Comput. Sci*. 2018, 132, 1351–1362.

[5] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using LSTM, RNN and CNN-sliding window model. *In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on,* pages 1643–1647. IEEE, 2017.

[6] Sezer, Omer Berat and Ozbayoglu, Ahmet Murat (2018). “Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach”. *In: Applied Soft Computing*.

[7] https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/

[8] https://www.datacamp.com/community/tutorials/lstm-python-stock-market

[9] Fama, Eugene F. (1965), “Random Walks in Stock Market Prices,” *Financial Analysts Journal*, 21 (5), 55–59.