Financial Technology: Assignment 4

Jan. 2021

B07703014 財金三 蔡承翰

(*i*) A general implementation of package *mplfinance* and *talib* gives the required figure.

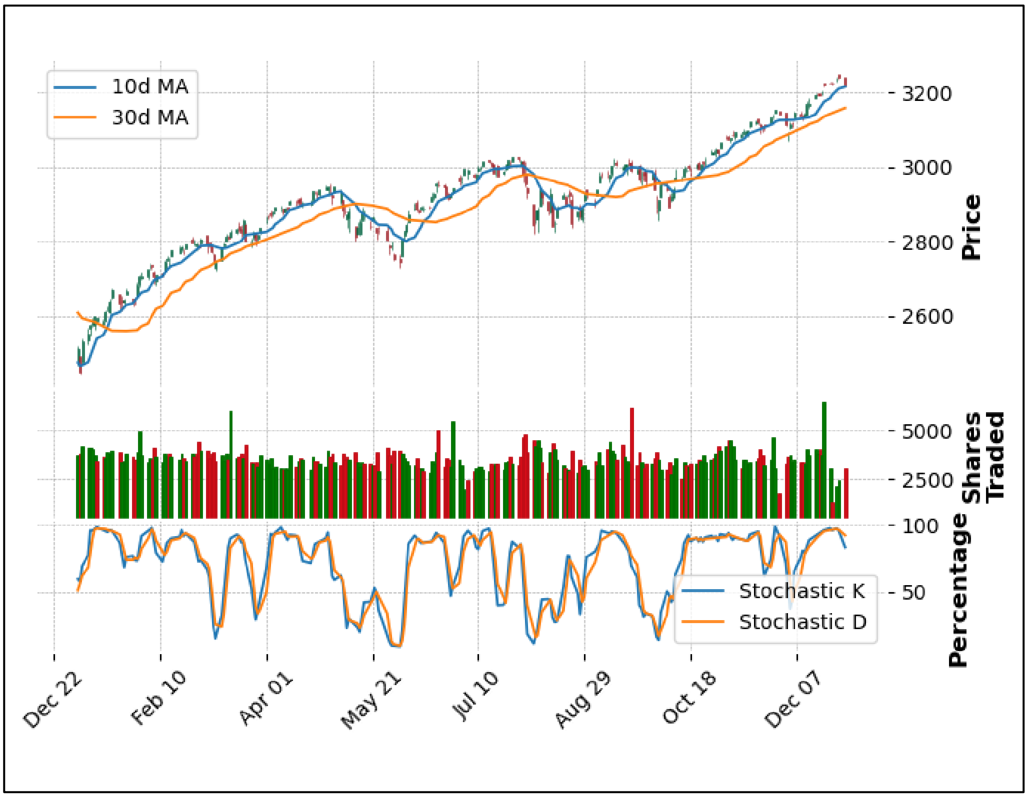

Figure 1

Figure 1 features several technical indicators for S&P 500 in 2019. A green candlestick means that the close price is higher than the open price on a specific day, while a red candlestick means the opposite. A green volume bar shows that the close price on a specific day is higher than the close price on a day before, while a green volume bar shows the opposite.

(*ii*) Following the instruction, we use all data available except *Adj Close* in *S_P.csv* to train the model. The reason why we abandon the feature is because that the *yfinance* package cannot give us such a feature when we crawl the S&P 500 data in 2020 later. The number of features that we use to train the model is 9.

(*iii*) We have chosen $batch\_size = 128$ in our model. As we use 9 features to do the training and prediction, $input\_dimension$ in the model should be 9.

(iv) We use *PyTorch* package to construct the model. The model is composed of an

RNN cell and a linear function before the final output. To make the model more feasible, assignments to several parameters in *__init__()* will substitute the RNN cell to an LSTM cell or a GRU cell. No matter which type of cell is created, we have made the hidden size be 32 and the number of layers be 3. In the training process, each mini-batch is composed of 128 days. Each day contains the values of the 9 features of last 30 days. The network then trains the model using these data.

(v) After training the RNN model, we have the conclusion of figure 2.

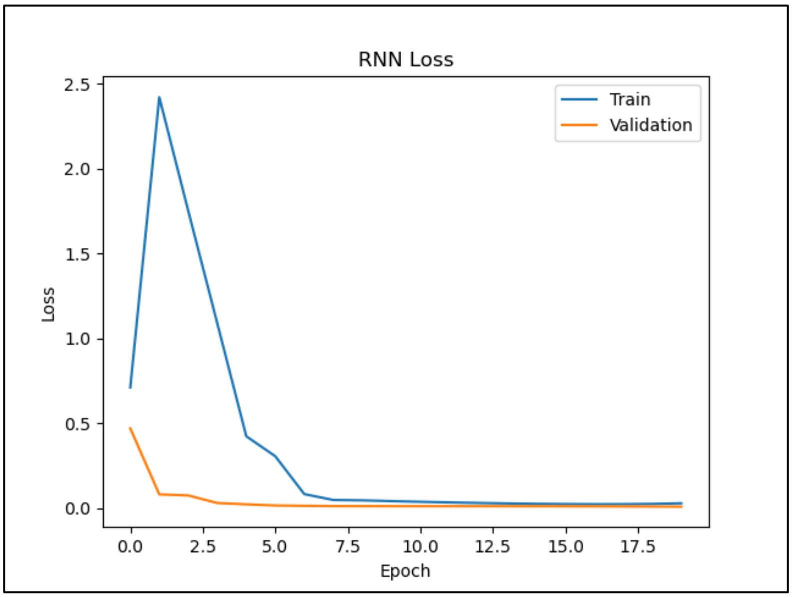

Figure 2

The RNN model achieved the train loss of 0.0278 and the validation loss of 0.0078 at the end. The prediction of the validation set is shown in figure 3.

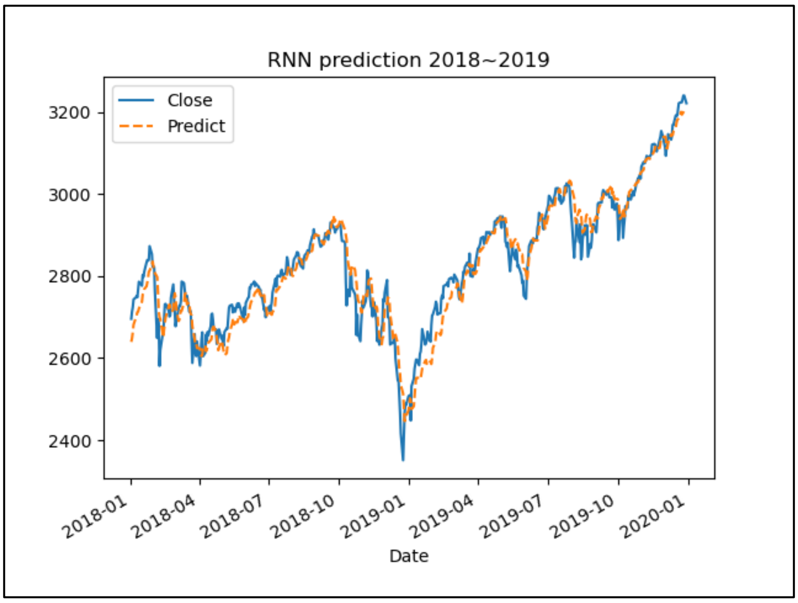

Figure 3

(vi) Substituting LSTM cell for RNN cell, we have the result shown in figure 4.

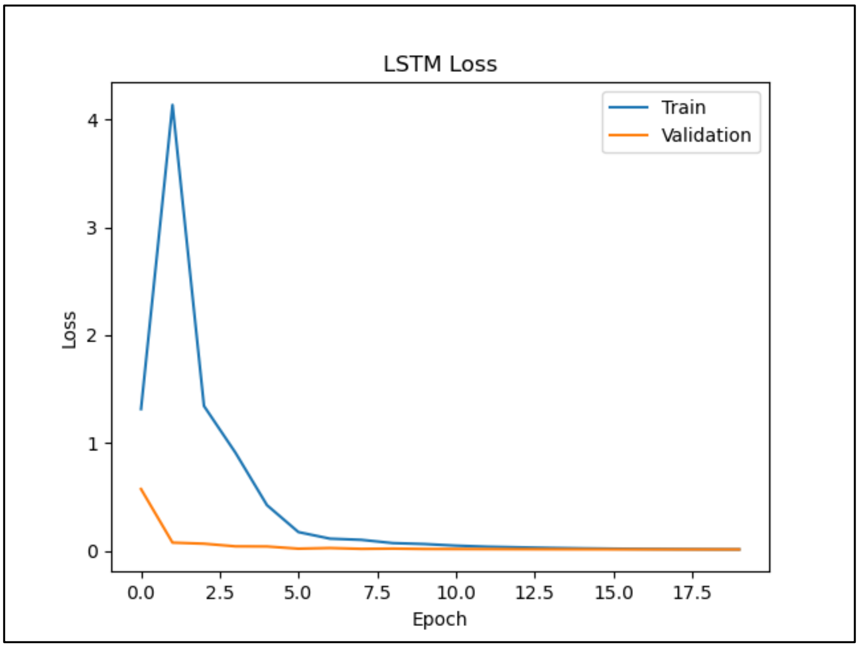

Figure 4

The LSTM model gave us the train loss of 0.0151 and the validation loss of 0.0148 finally. The prediction of the validation set is shown as figure 5.

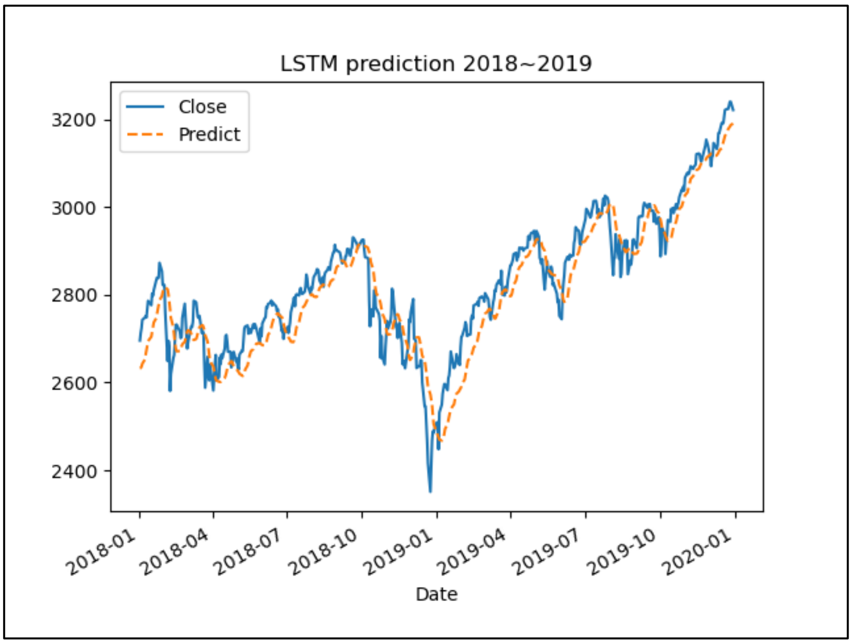

Figure 5

(vii) Substituting GRU cell for RNN cell, we have the learning curve shown in figure 6.

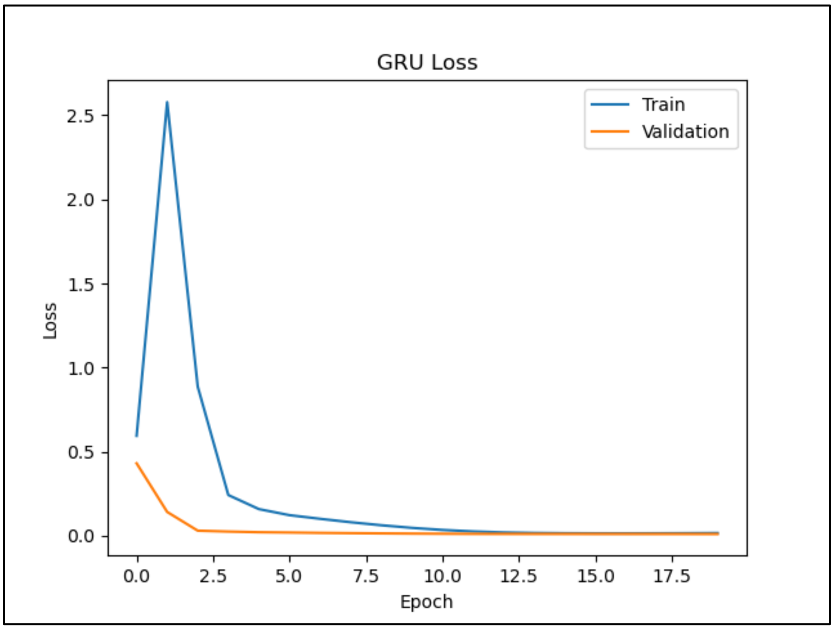

Figure 6

which yielded the train loss of 0.0156 and the validation loss of 0.0097. The prediction of the validation set is plotted in figure 7.

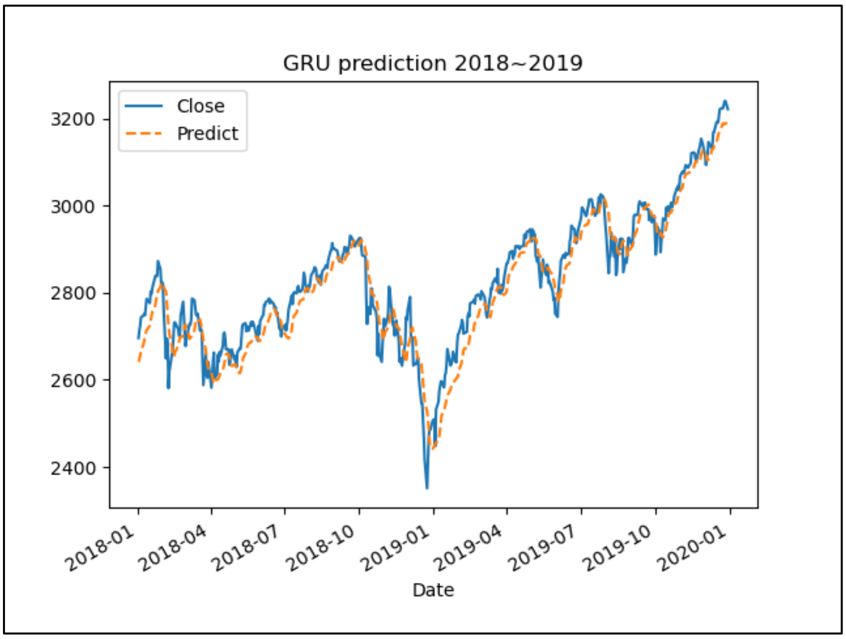

Figure 7

(viii) Among the three models we have applied to this task, RNN generates the smallest validation loss. Although LSTM and GRU are improvements of RNN, in this specific task it seems that RNN is the optimal model. Interestingly, we notice that in figure 3, 5, and 7, the prediction curves seem to be the lagged terms of the actual prices. This implies that the main factor for determining the S&P 500 index for a day is the historical prices. This is consistent to the famous *Efficient*

Market Hypothesis (EMH)[1]. We can infer from these figures that the weak form of EMH holds for S&P 500. That is, the price today is only affected by the historical price but not by any exogenous factor. Furthermore, the figures tend to consistent to the *Martingale Pricing Hypothesis*[2]. A martingale is defined as

$$E(y_t|\Omega_{t-1}) = y_{t-1}$$

In the stock case, y_t is the time series of stock prices, and Ω_{t-1} is all the price information before. The consistency to EMH and *Martingale Pricing Hypothesis* gives us the conclusion that despite the low loss, our model results may not make the S&P 500 profitable because that the index follows an unpredictable stochastic process.

(ix) The RNN, LSTM, GRU predictions for S&P 500 in 2020 are shown in figure 8, figure 9 and figure 10 respectively.

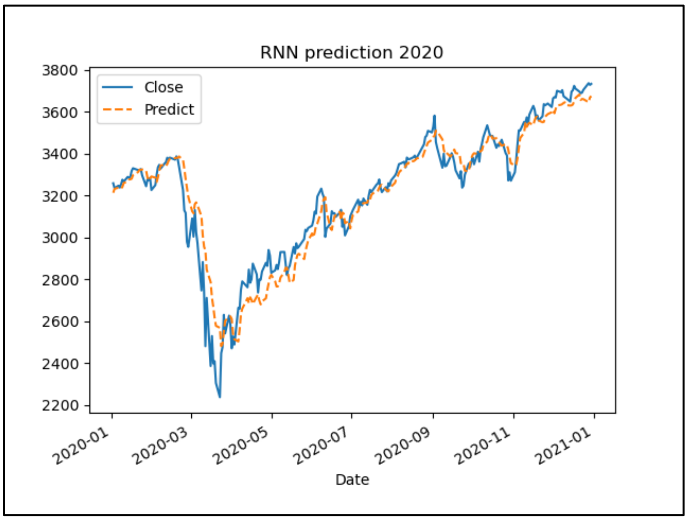

Figure 8

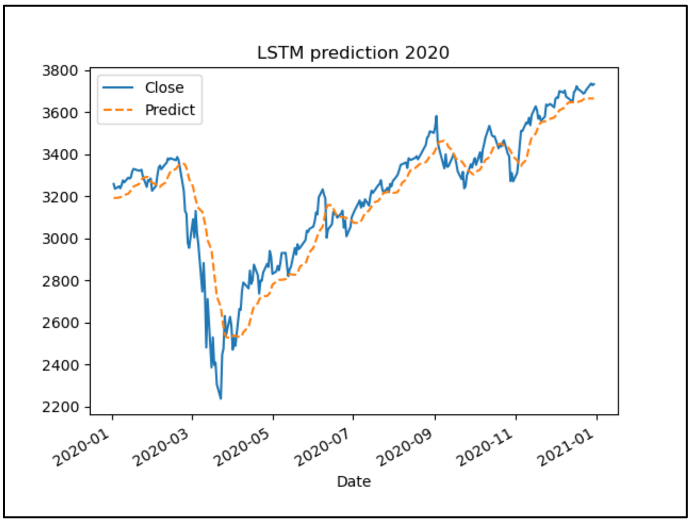

Figure 9

[1] https://www.investopedia.com/terms/e/efficientmarkethypothesis.asp
[2] https://en.wikipedia.org/wiki/Martingale_pricing

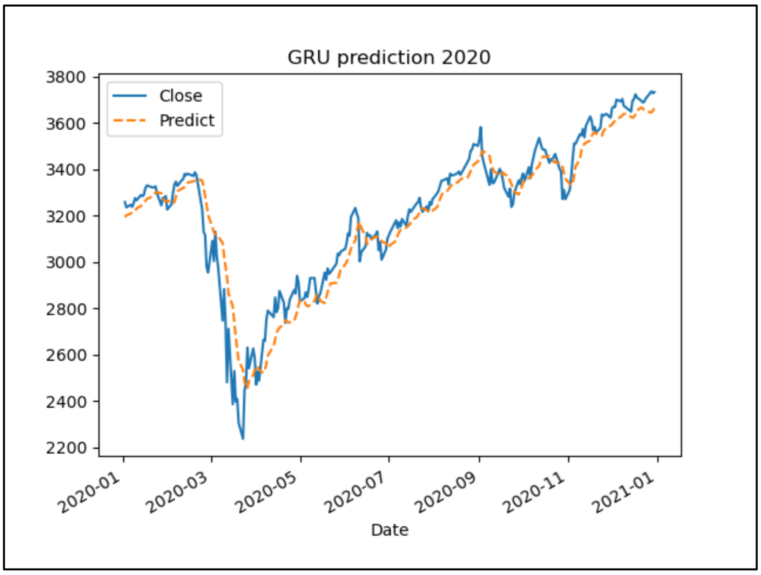

Figure 10

As we have discussed in part *(viii)*. We can see that our observation for figure 3, 5 and 7 holds for figure 8, 9 and 10 as well. Thus, we argue that the high predictability of the models is not because of finding the "pattern" of the index but because of the randomness of the change of the index. In fact, plenty of financial economics papers have shown that equity prices follow the unit root process (see Nelson and Plosser (1982), for example), which means that the first difference of the prices is weakly dependent. As a result, we do not consider that the randomness of S&P 500 index can be ruled out by our models.

(x) Though the randomness cannot be ruled out in our model, we propose that adding macroeconomics factor such as money supply or exchange rate will improve the performance. Several papers have shown that money supply can help predict stock returns (see Maskay and Chapman (2007), for example). Pícha (2017) has shown that the money supply can predict the stock returns on 6 months horizon. If we increase our time step to around 200 days and adding macroeconomics indicators such as money supply, exchange rate, inflation, income and unemployment rate in our features, the predictability of our model may be improved significantly.