Stock Price Forecasting Modelling

Chuanwu (Charles) Liu

December 16, 2019

1 Features

This task is a time series forecasting problem. Considering the significant data volume, I decide to use recurrent neural networks to solve this task. At time t, the input features are selected as the open, high, low and close prices and the trading volume at past times $t-N$, $t-N+1$, $\cdots$, $t-1$, t, where N is the lookback time. The volume is included, because it reflects the market's reaction to the prices and provides additional information. The derived features such as high/low are not included, because a neural network is able to approach these. Depending on the target feature, two models are considered:

1. The regression model, in which the target value is the close price at $t+1$, represented as P^c_{t+1}.
2. The classification model, in which the future close return $r_{t,t+1} = P^c_{t+1}/P_t - 1$ is calculated and the binary classification target is 1 if $r_{t,t+1}$ is positive or 0 if else.

2 Data Preprocessing

2.1 Time series

All valid trading dates are considered as consecutive dates. Non-trading dates including weekends and public holidays are ignored. Some tickers appear after Jan 2015, the dates before their first appearance is also ignored.

2.2 Missing values

For a selected ticker, if a data point is not found on a trading date after its first appearance, it is defined as a missing value. There are a number of methods have been considered for the handling missing values:

1. Discard samples with missing samples. This is the simplest way and and it does not introduce bias from missing value handling methods. However, in time series forecasting, the data removal will break the time consecutiveness.

2. Impute missing values with a value that has its meaning and distinct from all other values such as 0. This is acceptable, as neural networks will learn from the data that the value 0 means missing data and will start ignoring the value. However if missing value is included in the test set, we have to make sure the missing values are also included in the training set.

3. Impute missing values with the mean or median of each feature. Such a constant has small contribution to the loss function. However, it introduces bias if most of existing values are far from the mean or median.

4. Impute missing values with other predictive models such as linear interpolation, moving average, AIMA and Exponential Smoothing. However, this method requires extra effort to build predictive models for error handling. Subject to the model, the artificial patterns from the missing value handling model model may mislead the neural networks.

In my Jupyter notebook, four methods are implemented: (i) Remove missing values; (ii) Replace missing values with 0; (iii) Replace missing values with column means; and (iv) Linear interpolation. Their performances vary from ticker to ticker depending on the pattern of the signal and number of missing values. However, in this note, too reduce the number of free parameter, only the tickers with no or few missing values such as ANZ, CBA, NAB and BHP are presented.

2.3 Data Scaling

The price columns and the volume column have different scales so that data scaling is needed. There are two commonly used data scaling methods: the standard scaling and the MinMax scaling. The MinMax scaling is adopted in this task. In contrast, the standard scaling will result negative prices and volumes. The MinMax scaling is defined as

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

3 Close Price Regression

In this model I use the prices and volumes of past days, e.g. the past five days to predict the close price of the next day.

3.1 Error metric

We need an error metric to measure the difference between the model prediction and the test data. Because our error corresponds to the gain or loss of currency which is measured in liner scale. The Mean Absolute Error (MAE) is used as the error metric. MAE is defined as

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_{\text{pred}} - y_{\text{real}}|}{n}, \tag{2}$$

where y_{pred} and y_{real} are the predicted value and the real value respectively.

3.2 Baseline model

To evaluate of power of my model, I first build a simple model as the baseline. We expect our neural networks are able to overtake the simple model. In this simple model, I assume the close price remains same on the next day, i.e. $P^c_{t+1} = P^c_t$.

3.3 Recurrent Neural Networks

Recurrent Neural Network (RNN) is one the powerful tools for modelling sequential data including time series data. A Long Short-Term Memory (LSTM) network, which is a type of RNN is implement for this time series forecasting problem. The structure of my neural network contains:

1. Four LSTM layers.
2. Four Dropout layers with drop out rate 0.2
3. One Dense layer with activation function relu
4. One Dense layer without activation

The model is optimised with Adam optimizer. The loss function of the optimisation is MAE.

3.4 Training, validation and test samples

The data set is divided into training set (60%), validation set (20%) and test set (20%). The validation set is for the parameter tuning and test set is for the evaluation of model performance.

3.5 Hyper-Parameters

There are a number of hyper-parameters in this model:

- Lookback time, which is the number of past days that is used as input.
- Number of units in each layer. This is the dimension of the representation space (weight matrix) of each layer.
- Number of iteration epochs. One epoch represents a full iteration over all the training data.

I have completed a number of runs to tune the hyper-parameters. For example, to find out the number of iteration epochs. I have plotted the loss function for training and validation set as shown in Figure 1. We can see the model stop improving after ~ 100 epochs. Therefore 100 is adopted for epochs.

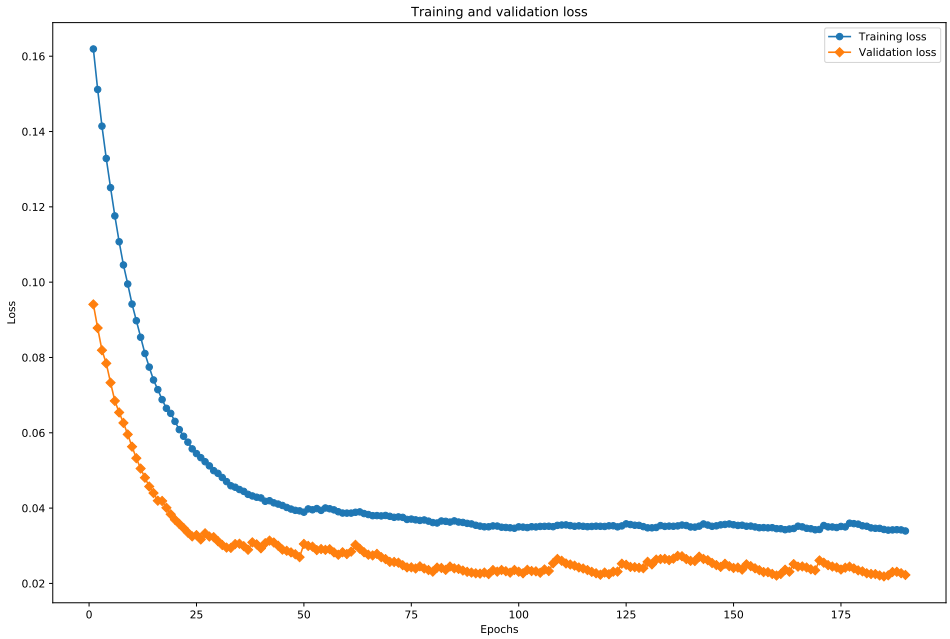

Figure 1: Loss of training and evaluation of 200 epochs for ANZ

Based on the quick parameters turning for a couple of example tickers. The adopted parameter set in this work is

Lookback time = 5

Units of layers = 64

Iteration epochs = 100

However, the parameter tuning is a time consuming work and is essential for a neural network. Comprehensive investigations are required for tuning the model.

3.6 Overfitting

I have added one dropout layer after each LSTM layer to avoid the overfitting. That is why we don't see overfitting in Figure 1.

3.7 Results

Figure 2 shows the results of model prediction compared with the test samples. We see the model does not perform well. This is because the model is not fully tuned.

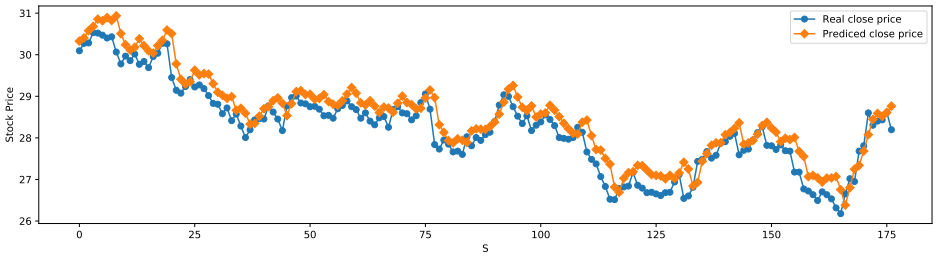

Figure 2: Loss of training and validation over 200 epochs for ANZ

Table 1: MAE loss of the regression

Ticker	Baseline loss	Model loss
ANZ	0.012	0.24
CBA	0.018	0.76
NAB	0.012	0.54
BHP	0.012	0.54

4 Binary Classification of future close return

In this approach, I first calculate the future close return $r_{t,t+1} = P^c_{t+1}/P_t - 1$ and the binary classification target is defined as

1 if $r_{t,t+1} > 0$
0 if $r_{t,t+1} \leq 0$

4.1 Error metric

The accuracy is adopted as the error metric. The accuracy is defined as

$$\text{Accuracy} = \frac{\text{Number of True Positive Predictions} + \text{Number of True Nagative Predictions}}{\text{Number of predictions}} \quad (3)$$

4.2 Baseline model

The baseline model for the binary classification is that I assume the future close return has the same sign as close return, i.e. $\text{sign}(r_{t,t+1}) = \text{sign}(r_{t-1,t})$.

4.3 The Model

The implemented network for this problem contains:

1. Two LSTM layers.

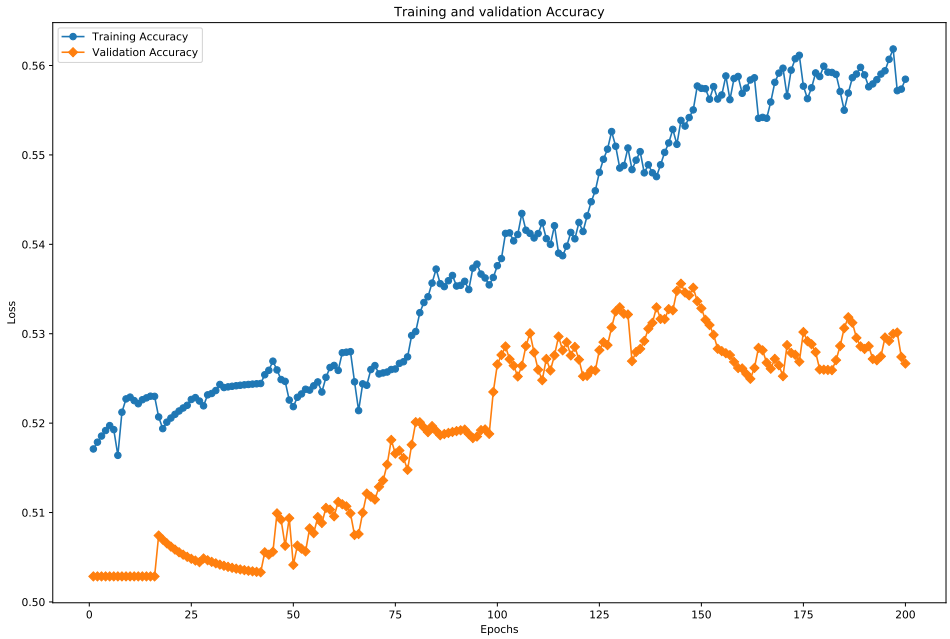

Figure 3: Classification accuracy of training and validation of 200 epochs for ANZ

2. Two Dropout layers with drop out rate 0.2
3. One Dense layer with sigmoid activation. The sigmoid activation is use to convert numeric values to probability.

The model is optimised with Adam optimizer. The loss function of the model is the binary crossentropy.

4.4 Hyper Parameters

The same set of hype parameters is adopted in this classification model. Figure shows the classification accuracy over 200 iteration epochs. We see that the accuracy reaches it maximum at ~ 100 and stop improving after that.

4.5 Results

The binary classification accuracy of example tickers is shown in Figure 2. We see that this current classification model is not able to predict the future close return, except for NAB, which has a

Table 2: Accuracy of the binary classification

Ticker	Baseline accuracy	Model accuracy
ANZ	48%	46%
CBA	45%	46%
NAB	44%	54%
BHP	57%	43%

5 Conclusion and Discussion

We see in both regression and classification approach, the RNN model does not overtake the simple baseline. This may be because of two reasons:

1. The models have not been well constructed and tuned. More comprehensive parameter tuning is required.
2. The stock prices of the past is not a good indicator of future prices.