Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 4, No. 1, 2025

A Method of Stock Price Forecasting Based on Recurrent Neural Network

Ganzhou Wu

School of science, Guangdong University of Petrochemical Technology, Maoming 525000, China

Abstract: It is selected the stock data of Kweichow Moutai for a certain period of time, uses python to model and analyze and predict the stock price, compares the predicted stock price with the real stock price, and then uses root mean square error, mean square error, and mean absolute error to evaluate the prediction model. RNN neural network can make good use of the nonlinear stock data and can memorize the effective information in the sequence data. Numerical experiments show that RNN neural network is a desirable stock forecasting method.

Keywords: Deep Learning, Stock Forecasting, RNN Neural Networks.

1. Introduction

At present, the stock return prediction methods appearing in the domestic and foreign stock markets can be roughly divided into the following three categories: (1) Fundamental analysis method and technical analysis method The fundamental analysis method is to analyze the current situation of a certain stock from the perspectives of economic cycle, industry, and enterprise quality, and formulate corresponding investment strategies based on its basic quality. The technical analysis method is a method that only judges the future trend of stock prices from market behavior. Commonly used technical analysis methods include trend line analysis, wave theory, K-line theory, angle analysis and other methods. This method cannot predict specific stock prices in the future. price and rate of return, but under certain conditions, the changing trend of stocks can be predicted. (2) Time series model method [1] Simply speaking, time series analysis is to observe and study time series, find its changing rules, and predict its future trend. However, when the external environment changes significantly, for example, when the national policy changes, there are often large deviations in the forecast results in the past historical data, which are only suitable for short-term forecasts, but cannot be used for long-term forecasts. Reliability is not high either. (3) Neural Network Analysis [2] For the most part, the upward and downward trend of stocks seems to be irregular. Stock market risks also come from many aspects. In order to solve the complex and nonlinear characteristics of stock data, we use neural networks with strong self-adaptation ability, strong generalization ability and good fault tolerance, and can affect stock prices. Various factors are analyzed and considered comprehensively. In addition, neural networks are essentially effective in finding relationships between data and using it to predict new data. It has been widely used in stock forecasting and has achieved certain results. Since the 21st century, the market economy has developed rapidly, which has also deeply affected the stock market. With the continuous development of the stock market, the amount of information contained in the stock exchange has also exploded, and the relationship between its direct and indirect factors can no longer be analyzed and summarized purely by manual. The rise of neural network algorithms and deep learning has added a lot of help to researchers' prediction of stock prices. Huang Liming and Chen Weizheng proposed a stock forecasting method based on multi-channel circular neural network and deep learning, and conducted experiments with Shanghai A-share stock prices and news data to prove its superiority [3]. Wang Ziyue used the neural circulation network to build a model to predict the different time intervals of one minute and ten minutes in the future to explore the law of minute-level stock prices [4]. Wang Jingwen used the LSTM model to predict the estimation of Zhongjin gold during the period from September 1, 2003 to December 29, 2017. The prediction research is mainly divided into four steps: first, use the LSTM model for training and testing forecast; Second, the root mean square error, loss function and prediction accuracy are statistically analyzed; Third, compare the prediction results of BP neural network structure; Fourth, summary evaluation [5]. Yuan Ruyi proposed that with the development of artificial intelligence technology today, AI technology should be applied to the financial industry, and the volatility prediction accuracy of financial assets can be improved by establishing a volatility prediction model [6]. Gray Orudnitsky and Larry Osborne compared the delayed neural network with the APdMA network, and the results showed that the APdMA network has better predictive performance [7]. In this paper, the RNN algorithm is established to lay a solid foundation for the follow-up research; Example validation analysis, select the real stock market data, modeling analysis, forecasting stock prices, and forecasting results analysis.

2. Recurrent Neural Network

2.1. The Principle of Recurrent Neural Networks

The structure of the traditional neural network is relatively simple, there are only input layer, hidden layer and output layer. The input layer accepts external input data, and finally outputs to the output layer after multiple calculations and processing by the hidden layer. Each neuron receives the output of the previous layer and produces a weighted sum, and passes the result to the next layer via an activation function. The weight parameters are obtained by the training algorithm, and the goal of the training algorithm is to adjust the weight so that the output of the neural network is as close as possible to the true value. Although traditional neural networks perform well in some tasks, their performance is relatively poor for more complex tasks. Therefore, in recent years, with the development of deep learning, deeper neural network structures such as convolutional neural networks and circular neural networks have begun to be widely used, and major breakthroughs have been made in different fields. Recurrent Neural Network (RNN) is a kind of neural network structure with memory function because it introduces directional loop compared with traditional neural network and can deal with the correlation problem between input values. Its main feature is to add a circular connection between the nodes of the neural network, so that the nodes can receive the output of the previous moment and take it as the input of the current moment. This circular connection can help networks "remember" past information and apply it to current calculations. In an RNN, each node has its own state, including a hidden state and an output state. The hidden state is used to remember the information in the past, and the output state is used to output the results at the current time. At each moment, RNN receives a new input, calculates a new output, and updates the state of the node. Each time, the previous output will be brought to the next hidden layer for training together, which reflects the relevance of learning. Through continuous iteration, RNN can process input sequences of any length and output corresponding sequences. Features learned by the RNN at each step of the network can be shared at all time steps, as shown in Figure 1:

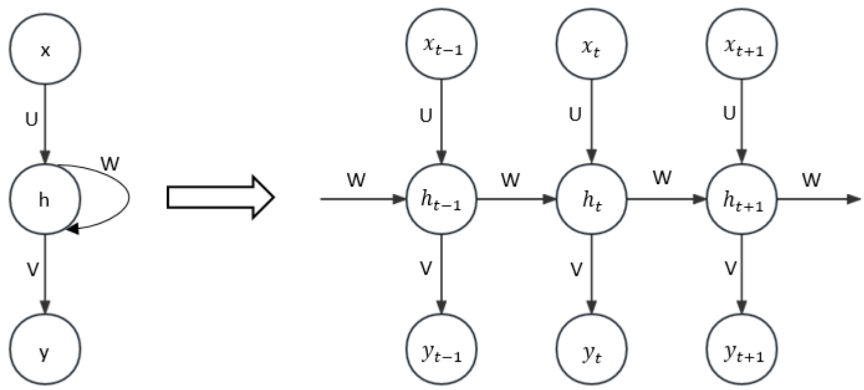

Figure 1. The structure of Recurrent Neural Network

We can take the schematic diagram on the left as the overall structure of the recurrent neural network, and the decomposition diagram of the recurrent neural network on the right. It can be seen that compared with traditional neural networks, circular neural networks have one more ring and the shared parameter W in this ring. We call this ring a recursive ring of circular neural networks, also called circular recursion, and W is called recursive parameter. As can be seen from the expanded graph, in the circular neural network, x represents the input, t is the current moment, t-1 is the previous moment, t+1 is the next moment, and y represents the output. Among them, the parameters W, U, and V are the same at every moment, which is the characteristic of parameter sharing in the recurrent neural network. Regardless of the chronological order, this weight does not actually change, which is also the biggest difference from the traditional neural network. difference.

BP error Backpropagation Through Time (BPTT) is usually used when training RNN, but BPTT cannot solve the long-term dependence problem, because BPTT will cause gradient disappearance or gradient explosion problem when training RNN, that is, the current output is related to the previous sequence, but generally does not exceed ten steps.

RNN is a kind of neural network with memory. By introducing circular connections in the network, the network can process the sequence data and use the context information to help predict the next output. Compared with other neural network structures, RNN is characterized by its ability to process arbitrary length of sequence data, and has dynamic input and output, the output of each time step will be affected by the input and output of the previous time step. Therefore, RNN is suitable for many sequence data processing tasks, such as speech recognition, natural language generation, machine translation and so on. At the same time, RNN also has some problems such as gradient disappearance and gradient explosion, which need to be solved by some skills. Here are the features of RNN:

(1) Recursive structure: RNN has a recursive structure, which allows information to be passed through the network and maintain state.

(2) Time dependence: RNN can process time series data and has time dependence.

(3) Memory characteristics: that is, the output of the previous moment can be used as the input of the next moment.

(4) Backpropagation: RNN can be trained using a backpropagation algorithm, and its weights can be optimized by a gradient descent method.

(5) Receive two parameters: the input at the current time and the output at the previous time.

(6) Parameter sharing, parameters W, U, V are the same in the loop in the network, ensuring that each step is doing the same thing.

3. Experimental Results

Kweichow Moutai (600519) stock from 2010.4.26 to 2020.4.26, the ten-year historical transaction data is obtained by a third-party interface, and the data is stored in the form of SH600519. csv. The following table is part of the sample data.

Table 1. Partial data

	Date	Open	Close	High	Low	volume	code
1	2010/4/26	10.801	9.246	11.237	9.223	107036.1	600519
2	2010/4/27	9.216	6.255	9.216	6.068	58234.48	600519
3	2010/4/28	5.542	5.639	6.594	4.79	26287.43	600519
4	2010/4/29	5.962	7.232	7.991	5.962	34501.2	600519
5	2010/4/30	5.113	3.31	5.113	2.349	85566.7	600519
6	2010/5/4	2.529	3.017	3.708	1.702	23975.16	600519
7	2010/5/5	2.386	4.633	4.971	2.011	33838.78	600519
8	2010/5/6	4.79	2.048	4.79	2.011	28240.34	600519
9	2010/5/7	1.034	4.047	4.317	0.658	31254.76	600519
10	2010/5/10	5.542	6.699	8.547	5.542	66192.58	600519
11	2010/5/11	8.239	6.789	8.547	6.165	33632.41	600519

Data cleansing There are some problems in the data that will affect stock forecasting. The most common cases are these three: First, a few garbled characters in the data will affect later operations. Converting the encoding format to utf-8 format can solve this problem. Second, in order to reduce the impact of data errors on model construction, first use software to write a program to delete data that is not a trading day. Third, it is necessary to artificially patch the wrong information caused by the lack of data, mainly using the transaction data of the previous day for data filling.

Division of training set and test set Since the company's stock price will fluctuate greatly after listing, after listing, we will delete some data after listing (usually data after 120 days), in order to make the training data and test data samples of Kweichow Moutai stock Relatively balanced, there are 2426 data in this article, and the opening price of the first 2126 days is selected as the training set, and the opening price of the next 300 days is used as the test set.

Add Data Labels The training set data of the first 2126 days is performed on the training set through a for loop, and the opening price of the training set for 60 consecutive days is extracted as the input feature x, and the data of the 61st day is used as a label. The for loop constructs a total of 2426-300-60 = 2066 group data. for the test set data of the last 300 days, use the for loop to iterate through the entire test set, extract the opening price of the test set for 60 consecutive days as the input feature x, and the data of the 61st day as the label. The for loop constructs a total of 300-60 = 240 sets of data. That is, the data is sorted into samples and labels: 60 timesteps and 1 output, each sample contains 60 time steps, corresponding to the label value of the next time step.

Standardization of data Before data analysis, it is usually necessary to collect a large number of different relevant indicators, and the nature, dimension, order of magnitude, availability and other characteristics of each indicator may be different, which makes it impossible for us to directly use it to analyze the characteristics and characteristics of the research object. rule. In the case of huge differences between different indicators, if the initial value of the indicator is used for analysis, the higher the indicator, the greater its role in the comprehensive analysis, and vice versa. For example, when evaluating the price index in different periods, the price increase of lower-priced vegetables and higher-priced home appliances can be included, but due to the large differences in their price levels, if their prices are directly used for analysis, it will The role of home appliances with higher price levels in the comprehensive index will be exaggerated, and the accuracy of the results is not high. In order to ensure the credibility of the results, it is necessary to transform the original index data so that different features have the same scale. Normalization refers to scaling data so that it falls into a small, specific interval. In some index processing of comparison and evaluation, we usually use this method to convert the unit limit of the data into a dimensionless pure number, so that various indicators can be easily weighted or compared.

In this paper, we use min-max normalization to find the inherent attributes of the training set, such as the maximum value and the minimum value, and then normalize the training set, and then normalize the test set, and the normalization is between 0 and 1. The normalization formula is as follows:

$$X' = (X-X_{min}) / (X_{max}-X_{min}) \tag{1}$$

The main principle of using RNN to model yield prediction is that the current output value of a sequence is not only affected by the output value of this layer, but also by the previous output value, that is, the nodes between hidden layers are fully connected, and the neural network will Memory the previous information, the input of the hidden layer at a certain moment includes not only the information of the input layer at that moment, but also the output of the hidden layer at the previous moment.

For the input data, the main work to be done is weighted and summed. After the input data is processed, it is consistent with the original data dimension, but the value will change with the weight and offset value. Among them, the initialization weights w0 and b0, w0 is a random value with a normal distribution, and b0 is uniformly set as an initial value of 0.

$$\text{hidden}=\sum_{i=1}^{N} x_i*w_i+b \tag{2}$$

After the data calculation in the previous step is completed, the result hidden is obtained, and then the activation function is used to process the data, here relu is selected as the activation function, and the negative value in the hidden is changed to 0, which can form the sparsity of the neural network and reduce the dependency between parameters. Finally, the result is used as the output of the next layer.

$$\text{out}=\sum_{i=1}^{N} \text{relu}(\text{hidden})*w_i+b \tag{3}$$

Where the mathematical formula for the relu function is:

$$\text{relu}(x)=\begin{cases}0(x<0)\\ x(x\geq 0)\end{cases} \tag{4}$$

For the output layer, there is no need to add an activation function, and the weighted sum can get the result, which represents the opening price of the next day.

$$pred=out*w_i+b \tag{5}$$

At this time, the predicted opening price pred is obtained, and the real opening price Y is calculated by calculating the mean value of the square difference to get loss.

$$loss=\frac{(pred\text{-}out)^2}{2} \tag{6}$$

After getting loss, the weight can be updated through the optimizer in TensorFlow. During the training process, the model and parameters are stored.

The parameters of the iteration loop during training are stored, and the results are as follows:

Table 2. Iterative loop parameter

Model: "sequential"		
Layer (type)	Output Shape	Param #
simple_rnn (SimpleRNN)	(None, 60, 80)	6560
dropout (Dropout)	(None, 60, 80)	0
simple_rnn_1 (SimpleRNN)	(None, 100)	18100
dropout_1 (Dropout)	(None, 100)	0
dense (Dense)	(None, 1)	101
Total params: 24,761 Trainable params: 24,761 Non-trainable params: 0		

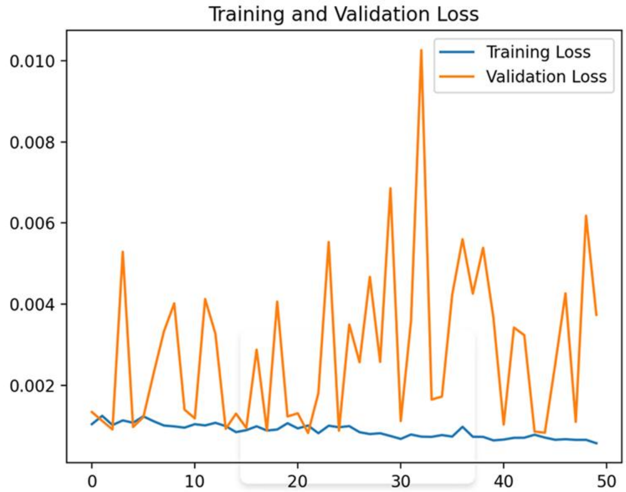

Figure 2. Forecast the training loss of the model

Figure 2 shows the training loss of the prediction model, where the orange curve represents the loss change trend of the test set data, and the blue curve represents the loss change trend of the training set data. The horizontal axis indicates the training batch, and the vertical axis indicates the loss value change. The loss value during model training is maintained between 0-0.02 and the change is relatively stable, while the actual stock loss value fluctuates between 0-0.01. From the training results of this model, it can be seen that the loss value reaches a very low level and the convergence interval is relatively stable, which has a good convergence effect.

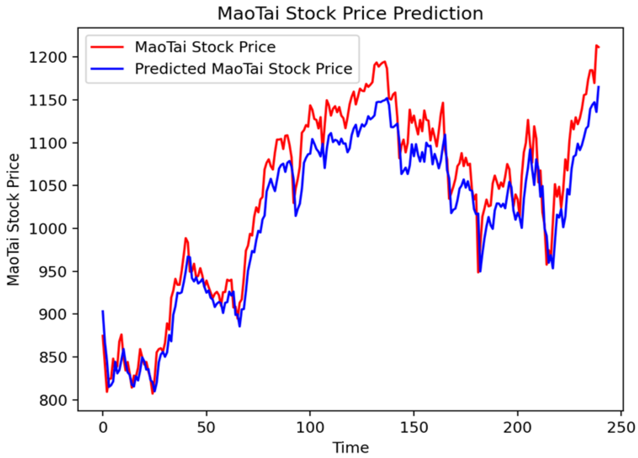

Figure 3. The Forecasting results

Figure 3 shows the prediction results of the training set, the blue line is the predicted price of the stock, and the red line is the actual price. From the figure, we can clearly know that the model can better simulate the trend of the stock forecast price and the trend of the actual stock price, so as to achieve higher forecasting efficiency.

MAPE (Mean Absolute Percentage Error) represents the average absolute percentage error, and the formula is as follows:

$$MAPE=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i\text{-}y_i}{y_i}\right| \tag{7}$$

It can be calculated by substituting formula 7,

APE=28.473641

This shows that the prediction effect is very good.

4. Summary

The numerical results show that the RNN neural network can make good use of the non-linear stock data and can memorize the effective information in the sequence data, which shows that the RNN neural network is a desirable stock forecasting method.

References

[1] Liu Changkun. Research on stock prediction based on deep learning [D]. Beijing University of Technology. 2018.

[2] Wang Litong, Xue Tengteng, Wang Huimin, Liu Zhen. Research on Stock Index Price Forecasting Based on Recurrent Neural Network [J]. Journal of Zhejiang University of Technology, 2019, 47 (02): 186-191.

[3] Huang Liming, Chen Weizheng, Yan Hongfei, Chen Chong. Stock Forecasting Method Based on Recurrent Neural Network and Deep Learning [J]. Journal of Guangxi Normal University (Natural Science Edition), 2019, 37 (01): 13-22.

[4] Wang Ziyue. Recurrent Neural Network Stock Forecasting [J]. Computer Knowledge and Technology, 2018, 14 (22): 171-172.

[5] Wang Jingwen. Empirical Analysis of Stock Prices Based on Recurrent Neural Network [D]. Yunnan University. 2020.01 .

[6] Yuan Ruyi. Stock Forecast Analysis Based on Deep Learning [J]. China Collective Economy, 2021, (24): 105-106.

[7] Gary Orudnitski & Larry Osburn.Forecasting S&P arid Gold Futures Prices: An Application of Neural Networks Iii. [J]. The Journal of Futures Markets, 1993, 6 (3): 631-643.