Chapter 32
Stock Price Forecast Based on LSTM Neural Network

Qiang Jiang, Chenglin Tang, Chen Chen, Xin Wang and Qing Huang

32.1 Introduction

Artificial neural network development history:

In 1943, French psychologists W. S. McCuloch and W. A. Pitts merged the results of biophysics and mathematics. Based on the analysis and synthesis of the basic features of neurons, the most primitive and elementary M-P model of neuronal synaptic models was proposed. This was the first one since W. James used collective parallel computing structure to describe the artificial neural network and network working mode, and proved that the MP model can perform any finite logic operation. Creating a new scientific ANN in the history of human science and technology [15]. In 1948, John Von Neumann compared the relationship between human brain structure and instruction storage computer, and proposed a simple neuron to construct the network structure of self-regenerative automata [18]. In 1949, psychologist DO Hebb proposed the notion of neuron populations, synapses and return circuits, arguing that information can be stored on neuronal junctions. According to the theory of conditioned reflexology in psychology, he studied the appropriate learning methods in ANN, and discussed the changing rules of the connection strength between nerve cells, summarized into the famous Hebb learning rule: if two neurons are excited and activated, Synaptic connection will be enhanced [7]. In 1958, Rosenblatt proposed a "perception machine" model with learning ability to complete the transition from a single neuron to a three-layer neural network. The prototype of the three perceptrons consists of a sensing layer S, a connecting layer A and a reactive layer R. Since the connecting right from the sensing layer S to the connecting layer A is fixed, the connection right from the connecting layer A to the reaction layer R has the ability to change due to learning. Therefore, it is only a single-layer neural network with only input and output layers. Due to the simple concept of the perceptron, it was put

Q. Jiang · C. Tang (✉) · C. Chen · X. Wang · Q. Huang
Business School, Sichuan University, Chengdu 610064, People's Republic of China
e-mail: 840079520@qq.com

J. Xu et al. (eds.), *Proceedings of the Twelfth International Conference on Management Science and Engineering Management*, Lecture Notes on Multidisciplinary Industrial Engineering, https://doi.org/10.1007/978-3-319-93351-1_32

emphasis on for the first time. However, Minsky and Papert finally proved mathematically that the perceptron can not complete complex logic functions [16]. In 1960, B. Widrow and ME. Hoff proposed an adaptive linear element "Adaline" network, which is a monolayer feedforward perception machine model proposed on the basis of the study of adaptive brain learning system at that time. The mean square error minimization algorithm it used and the perception machine error correction algorithm are the same in form, the difference lies only in the change of the threshold sign. Because Adaline's linearity and adaptability and Least Mean Square (LMS) algorithms have rigorous mathematical foundations, Adaline has become a powerful tool for adaptive signal processing [20]. In 1961, ER. Caianiello published his theoretical work on neural network mathematics, proposed neuron equations, used Boolean algebra to simulate the kinetic process, analyzed and developed a theoretical model of cellular finite automaton [2]. In 1969, American artificial intelligence experts Minsky and Papert published Perceptrons, proving that single-layer neural networks can not even solve simple arithmetic problems such as XOR and can not train many patterns which have been found [13]. In 1974, when Paul Werbos of Harvard University studied social sciences, he discovered the mathematical principle of the BP algorithm [19]. In 1977, Amari studied the mathematical theory of neural networks [1]. Based on the idea that human brain automatically organizes its Recognition Code, Grossberg and Gail Garpenter proposed the ART (Adaptive Resonance Theoty) mapping theory. The dynamic characteristics of the network was described by first-order differential equations. In 1981, Kohonen proposed an unsupervised self-organizing learning neural network model, which is called SOM (Seh-Organizing fenture Map), which developed self-organization mapping [12]. In 1982, Fukushima added a hidden layer to the single-layer perceptron, achieved self-organized learning through inhibitory feedback and excitatory feed-forward, so that multi-layer perceptrons to achieve the association learning and pattern classification recognition [4]. In 1982, the biophysicist Hopfield of the California Institute of Technology adopted the fully interconnected neural network model, applied the concept of energy function, succeeded in solving the problem of traveling salesman optimal path (TSP) for classical artificial intelligence problems that digital electronic computers are not good at solving. This is a major breakthrough in the history of ANN research, has aroused great concern around the world, but also set off a boom in neural network research [9]. In 1983, TJ Sejnowski and GE Hinton put forward the concept of "hidden element", introduced Boltzmann machines for massively parallel processing, which changed the unit connection rights by using parallel distribution of multi-layer neural networks and overcome the limitation of single-layer network. Laid the foundation for neural networks to enter the field of nonlinear processing [3]. In 1986, Rumelhart and his colleagues published the famous Parallel Distributed Processing monographs and proposed a back propagation learning algorithm for multi-layer feed forward networks, or BP algorithm for short. The algorithm amended the connection rights between layers from the back and negated the wrong conclusion of 1969 to multi-layer networks. Since then until now, BP algorithm has become the most widely used, the most studied and the fastest growing algorithm [17]. From June 21 to June 24, 1987, the Institute of Electrical and Electronics

Engineers (IEEE) held its first World Conference on Neural Networks in San Diego, USA. The International Neural Network Society (INNS) was also born, it marks that ANN research has spread all over the world. From 1988 to now, more and more academic activities, research institutes, monographs and other academic journals with keen insight had published a large number of ANN research articles. Heaton et al. proposed a deep learning level decision model for financial forecasting and classification. Proved that deep learning has the potential to improve predictive performance in traditional applications [6]. Shusuke Kobayashi and Susumu Shirayama proposed a new method of time series prediction using multiple deep learners combined with Bayesian networks. Compared with the traditional method of using a single deep learner, the F value and accuracy was improved [11].

Recurrent Neural Networks (RNN) is a kind of feedback neural network. Tracing the source, RNN was first inspired by the Hopfield network variants [9]. The Hopfiled network was proposed by John Hopfield in 1982 as a network structure, there was a feedback connection inside such network, that can handle time dependencies in the signal. In 1986, Michael Jordan borrowed the idea of Hopfiled network, for the first time, introduced circular connections in neural networks [10]. In 1990, Jeffrey Elman formally proposed the RNN model on the basis of Jordan's research, but at that time RNN was also called Simple Recurrent Network (SRN) [14]. Due to the introduction of the loop, RNN has the ability to have limited short-term memory, which has some advantages in the mining of sequence data. It is widely used in natural language processing, speech recognition, handwriting recognition, machine translation and other fields.

LSTM (Long Short-Term Memory) was proposed by Hochreiter and Schmidhuber (1997) [8] and has been recently improved and promoted by Alex Graves [5]. LSTM has achieved considerable success on many issues and is widely used. Due to the unique design structure, LSTM is particularly suitable for dealing with the tasks that its timing intervals and delays are very long, and its performance is excellent. As a non-linear model, LSTM is very suitable for constructing larger deep neural networks.

The LSTM system can also learn to control robots, analyze images, summary documents, recognise the videos and handwriting, run chat bots, predict disease and click rates, compose music, and much more.

First of all, this paper theoretically introduces the internal structure principle of RNN and LSTM, builds the entire neural network structure step by step. The analysis confirmed that both of them are excellent predictors of time series data. At the same time, due to the design of LSTM gate unit control, the effective storage and filtering of memory makes it more excellent in long-term memory and more prominent in predicting time series data.

Secondly, this paper uses daily market data from Shanghai Composite Index and Dow Jones Index, and builds up the two neural networks in turn and trains them to learn and forecast. Compared with the real value, the prediction deviation is calculated, which proves that the LSTM model is more accurate than the traditional RNN model.

Finally, by comparing the forecasting effect of the Shanghai Composite Index and the Dow Jones Index, the empirical study finds that the adaptability of China's stock index to the neural network model does not have a good adaptability to the US index. It is possible that the Chinese stock index has other hidden parameters that are not consider. So the predictive effect of neural network model is not as good as the US index.

Although the neural network model is feasible for the prediction of time series, it still needs to be improved. The three-layer neural network constructed in this experiment is relatively simple and the prediction effect is ideal. However, the model itself can be further optimized, at the same time, the parameters used in the model can also be optimized to make it more predictive.

32.2 Basic Theory

The purpose of RNN is to process sequence data. In the traditional neural network model, from the input layer to the hidden layer to the output layer, the layers are fully connected with each other, and the nodes between each layer are connectionless. However, this common neural network is ineffective for many problems. The reason why RNN is called recurrent neural network, that is, the current output of a sequence is also related to the previous output. Specifically, the network memorizes the previous information and applies it to the calculation of the current output, that is, the nodes between the hidden layers are no longer connectless but connected, and the input of the hidden layer includes not only the output of the input layer also includes the output of the hidden layer at the last moment. In theory, RNNs can process sequence data of any length. However, in practice, in order to reduce the complexity, it is often assumed that the current state is only related to the previous few states. Traditional RNN mostly uses Back Propagation Time (BPTT) algorithm. The disadvantage of this algorithm is that, as time goes by and the number of network layers increases, it can cause problems such as gradient vanishing or gradient explosion. It makes the traditional RNN difficult to deal with the problem of long-term dependence in practice.

LSTM was originally designed to solve the long-term dependency problem in neural networks and to remember long-term information as the default behaviors of neural networks, rather than requiring much effort to learn. The LSTM model is a kind of recurrent neural network (RNN). It proposes an improvement for the existing gradient vanishing problem of RNN model, replaces the hidden nodes in the original RNN model with a memory unit. The LSTM model uses the accumulated linear form to process the information of the sequence data to avoid the problem of gradient vanishing and to learn long-period information, thus overcoming the shortcomings of the RNN model. LSTM can learn long-term and short-term time-dependent information. Because of neural networks containing time memory elements, LSTM is suitable for processing and predicting interval and delay events in time series. The

LSTM model is a commonly used deep learning model for processing time series data.

The following is the brief introduction to the recurrent neural network and LSTM structure principle.

32.2.1 Single-Layer Network

$$y = f\,(Wx + b)\,.$$

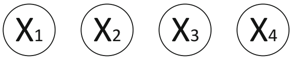
This is a basic monolayer neural network model, the input is x, the output y is transformed by transforming $Wx + b$ and activated by the function f.

32.2.2 Classic RNN Structure

Here are some classic recurrent neural network structures.

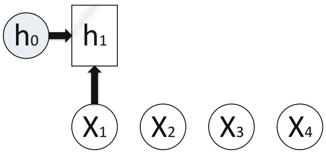
(1) (N vs. N)

In practice, we will encounter a lot of sequence-shaped data:



In order to model the sequence problem, RNN introduces the concept of hidden state h, which extracts features from sequence-shaped data and then converts it to output.

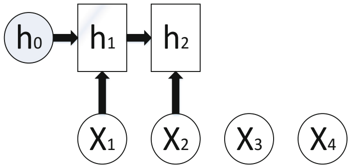
$$h_1 = \; f\,(Ux_1 + Wh_0 + b)\,.$$



The meaning of the symbols in the diagram is:

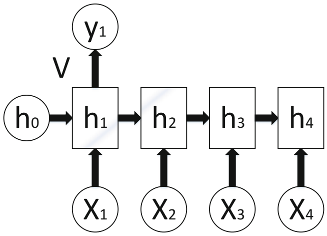
(1) Circles or squares represent vectors.

(2) An arrow indicates a transformation of the vector. As shown above h_0 and x_1, respectively, there are arrows connected, it means that the h_0 and x_1 made a transformation.



$$h_2 = f(Ux_2 + Wh_1 + b).$$

In calculation, the parameters U, W, b used at each step are the same, that is to say, the parameters of each step are shared, which is an important feature of RNN. Calculate the remaining ones (using the same parameters U, W, b).



$$y_1 = Soft\ max(Vh_1 + c).$$

The rest of the output is done similarly (using the same parameters V and C as y_1).

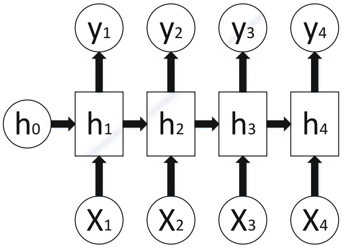
This is the most classic RNN structure. Its input is x_1, x_2,...x_n, the output is y_1, y_2, ...y_n, that is, the length of the input and output sequences must be equal.

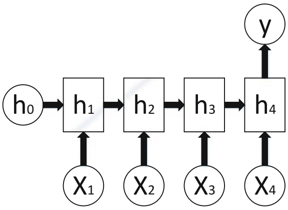
(2) (N vs. 1)

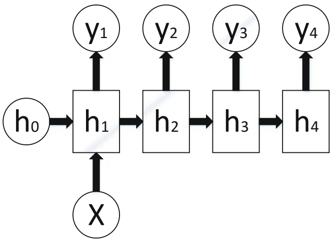
$$y = Soft\ max(Vh_4 + c).$$

This structure is roughly the same with the classic structure, just at the end of the output, only the last state is output. This structure is usually used to deal with the classification of the sequence.

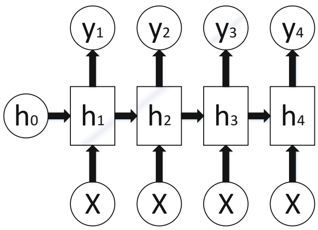
(3) (1 vs. N)







Input is only one vector that x. According to the number of outputs, build the hidden layer, and then make a conversion to the hidden layer, finally output the y_1, $y_2, \ldots y_n$.

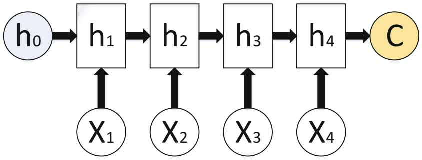


This model also has another structure. That is, use x as the input for each stage, and then make a transformation to the hidden layer, finally output the $y_1, y_2, \ldots y_n$.

(4) (N vs. M)

The sequence lengths of the input and output structure are not the same, but also not 1 at the same time, this structure is also called Encoder-Decoder model, also can be called Seq2Seq model. The following is a brief introduction of the model.

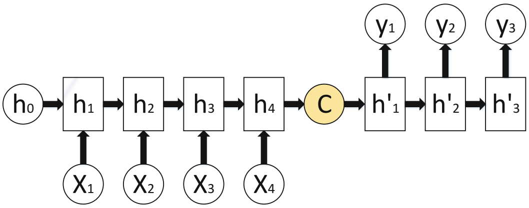
In the beginning, the Encoder-Decoder structure encodes the input data into a context vector C.



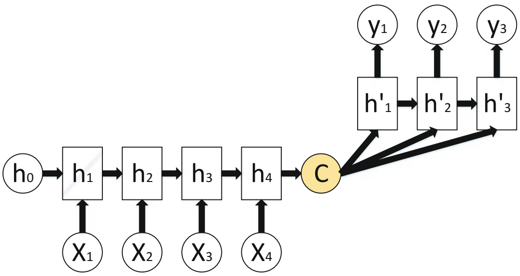
$$\begin{aligned} c &= h_4, \\ c &= q\left(h_4\right), \\ c &= q\left(h_1, h_2, h_3, h_4\right). \end{aligned}$$

There are many ways to synthesize C, you can assign the last hidden state to C, you can make some changes to the last hidden state then assign to C, and you can also assign some changes of all hidden states to C.

After getting C, it will be decoded by another RNN network, which is called Decoder. Enter C into the decoder as the previous initial state $h0$, then use another network to transform and output the result.



Similar to the previous approach, C can be used as the input for each step of the decoder while decoding.

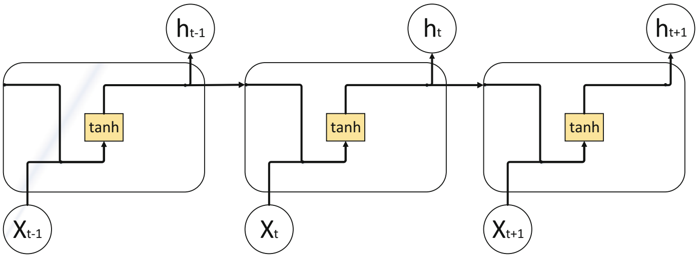


These are some classic RNN models.

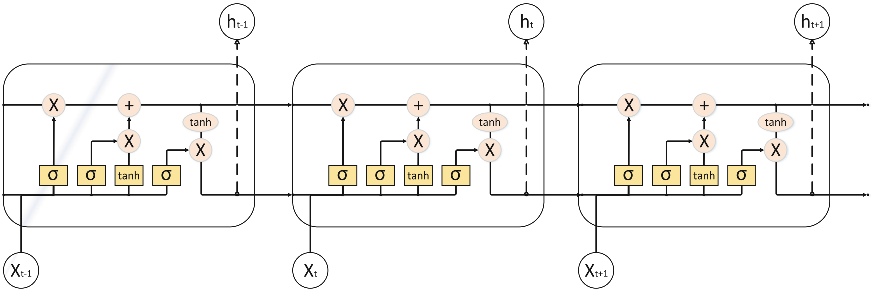
32.2.3 LSTM Network

In the process of information transmission, due to the problems of gradient vanishing or gradient explosion, if the memory of the past is far away from now, the information of the past may have been forgotten. LSTM is based on the original to add a state, that memory unit C, it can reasonably keep long-term status, the memory information from the initial position of the sequence, passed to the end of the sequence.

Standard RNN internal structure, assuming that the module has only a very simple structure, such as a tanh layer:



LSTM is the same external structure, but the module has a different internal structure.



C is a memory unit, used to save long-term information.

LSTM designed four door switches for effective control of this memory unit C.

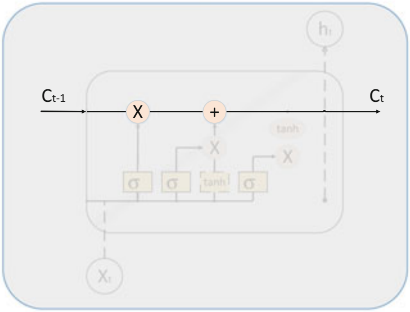
(1) Forgotten Gate

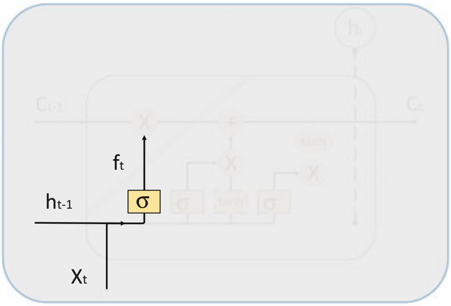
$$f_t = \sigma\left(w_f{}^T * h_{t-1} + U_f{}^T * x_t + b_f\right).$$

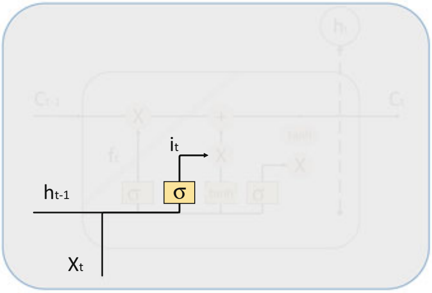
σ represents the activation function, which is usually sigmoid here. $w_f{}^T$ denotes the forgetting gate weight matrix, $U_f{}^T$ is the weight matrix between the input layer and the hidden layer of the forgotten gate, b_f denotes the biases of the forgotten gate, where subscript f is the first letter of "forget", in order to enhance readability only. The output f_t value is closer to 1, the more information is retained. Conversely, closer to 0, the less reservation is reserved.

(2) Input Gate

$$i_t = \sigma\left(w_i{}^T * h_{t-1} + U_i{}^T * x_t + b_i\right).$$







The formula is almost exactly the same as forgetting the door, and sigmoid is also used to activate the function σ.

(3) Candidate Gate

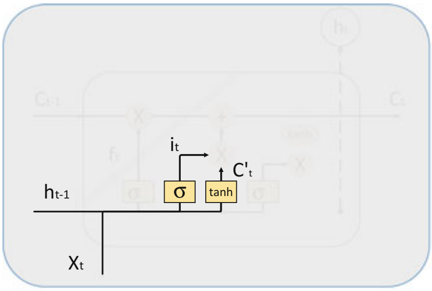
$$c_t' = tanh\left(w_c{}^T * h_{t-1} + U_c{}^T * x_t + b_c\right).$$

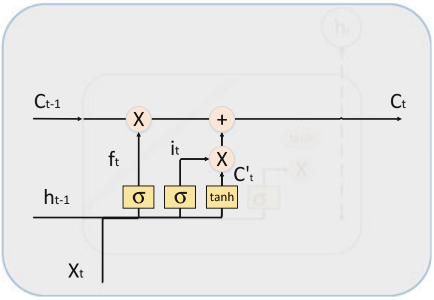
The activation function is replaced by tanh, which normalizes the output to between -1 and 1.

$$c_t = f_t * c_{t-1+}c_t' * i_t.$$

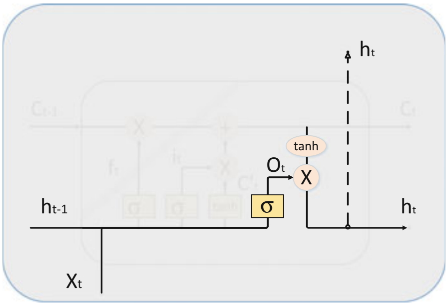
The combination of input gate and candidate gate generates a new memory unit. Due to the Forgotten Gate, it can control how much the previous information is saved. Due to the input gate, it again avoids the memory of the currently unimportant content.

(4) Output Gate





After the internal memory state update is completed, the following will decide whether to output.



$$O_t = \sigma\left(w_o{}^T * h_{t-1} + U_o{}^T * x_t + b_o\right),$$
$$h_t = O_t * tanh\,(C_t)\,.$$

Activation function here is still used sigmoid. h_t is the current output, C_t is the updated memory unit.

So far, we have analyzed the internal standard design of the LSTM network. Many machine learning specialists had also improved on the basis of LSTM and developed many simple but equally effective algorithms.

32.3 Empirical Results

In this experiment, the data we used is the Shanghai Composite Index and the Dow Jones Index daily data, the time period is from October 31, 2007 to October 31, 2017. The source of the data used for this time is from the Tong Daxin software, the way of the data restoration of rights is complex before right, and the reply date is December 8, 2017. The opening price, the closing price, the highest price, the lowest price of the daily data as the independent variables, the closing price of the next day as the dependent variable. The first eight years of the data as a training set, the next two years of the data as a test set. At first we train the training set, and then use the test set to predict, observe the fitting effect and test the model's predictive ability.

In this experiment, we use the Shanghai Composite Index and the Dow Jones Index daily data, respectively, to establish RNN model and LSTM model. We use a three-layer neural network, input layer, hidden layer and output layer. The number of hidden layer nodes tentatively scheduled for 10.

The fitting effect is shown in the following figure: blue line is the exact value and red line is the fitted value (Fig. 32.1).

The observation shows that the fitting of the Dow Jones Index using the LSTM model shows that the fitting effect is ideal and the basic trends and transitions are fitted out (Fig. 32.2).

From the observation, we can see that the Dow Jones Index fitted by using the RNN model has a larger fluctuation than the LSTM, especially in some turning points (Fig. 32.3).

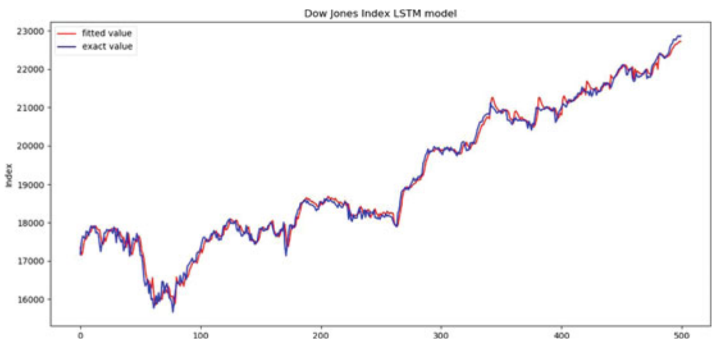


Fig. 32.1 Dow Jones Index LSTM model

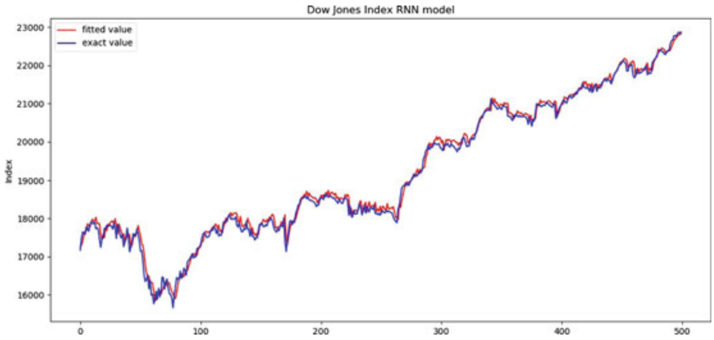


Fig. 32.2 Dow Jones Index RNN model

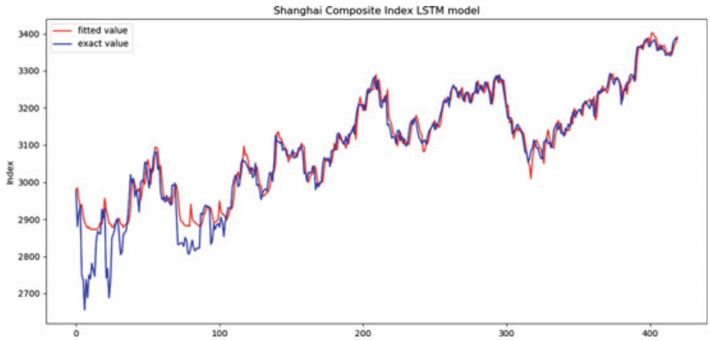


Fig. 32.3 Shanghai Composite Index LSTM model

Fig. 32.4 Shanghai Composite Index RNN model

Observation can be found, the use of LSTM model for fitting the Shanghai Composite Index, except for the beginning of a period of violent fluctuations did not fit in place, the latter fitting is basically the same (Fig. 32.4).

The observation shows that the Shanghai Composite Index, which is fitted by using the RNN model, although the previous section has been roughly fitted, but overall

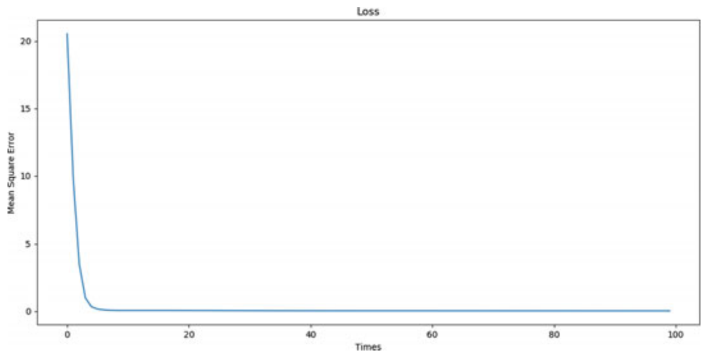


Fig. 32.5 The mean square error

Table 32.1 The degree of deviation

	LSTM	RNN
Shanghai Composite Index	0. 007554	0. 007920
Dow Jones Index	0. 006803	0. 008023

it seems, the volatility is rather violent, with some large continuous fluctuations at some turning points.

The figure is the mean square error changes with the number of iterations in the training process (Fig. 32.5):

In this training process, the number of iterations is 100. As can be seen from the figure, in the training process, with the increase of the number of iterations, mean square error began to rapidly decrease, and finally remained stable and consistent.

The following table is the degree of deviation between the predicted value and the true value (Table 32.1):

The degree of deviation is calculated as "abs ((predicted value—true value)/true value)" and then averaged. As can be seen from the above table, the degree of deviation of RNN is slightly larger than that of LSTM. For the prediction of the Dow Jones Index, the degree of deviation is less than the prediction for the Shanghai Composite Index.

32.4 Conclusion

Through this experiment, we can find that, for the neural network model compared with the RNN model, the LSTM model fits the data well. As for the different Indexs, the fitting effect of the Dow Jones Index is even more excellent. The Shanghai Composite Index fluctuate greatly and the fitting difficulty is relatively high. This is because the Chinese stock market is essentially different from the European and

American stock markets. The Chinese stock market belongs to the policy market and the European and American stock markets belong to the value market. Most of the data of the European and American stock indexes depend on the value of the stock or the company itself, so there are few ups and downs and the black swan phenomenon. Therefore, the model is more accurately predicted. But the ups and downs in China's stock market depend largely on policy and good news, and most of China's stock market participants are ordinary citizens, unlike institutional investors in European and US stock markets. Ordinary people are more herding, not rational. Therefore, in fact, in order to accurately predict China's stock index, more hidden parameters should be added into the model, such as the influence of policy information, etc. However, this parameter is extremely difficult to quantify, so the forecast for China's stock index so far there is still no better solution. The application of machine learning method in China market still needs more efforts and tests.

With the development of deep learning, more and more algorithms have been invented by scientists. Many simpler but equally superior algorithms have been invented. We can also learn these algorithms and apply them in different fields. With the popularization of artificial intelligence, artificial intelligence deepens people's life more and more. We are fortunate enough to catch up with this beautiful era, the era when artificial intelligence is flourishing and the climax of artificial intelligence. We must grasp the trend and learn artificial intelligence, give our own strength to artificial intelligence.

Acknowledgements This research accepts the following grants: Chengdu Soft Science Research Fund (Project No.: 2015-RK00-00087-ZF; 2015-RK00-00257-ZF); Research Center Fund of Sichuan County Economic Development, Sichuan Provincial Key Research Center for Social Sciences (Project No.: XYJJ1506); Basic Scientific Research Fund of Sichuan University (Project No.: skqx2015-zx04; skzx2015-sb68; skyb201402); Humanities and Social Sciences Fund of Chinese Ministry of Education (Project No. 14YJC790053).

References

1. Amari S (1977) Dynamics of pattern formation in lateral-inhibition type neural fields. Biol Cybern 27(2):77–87
2. Caianiello ER (1961) Outline of a theory of thought-processes and thinking machines. J Theor Biol 1(2):204
3. Fahlman SE, Hinton GE, Sejnowski TJ (1983) Massively parallel architectures for al: Netl, thistle, and boltzmann machines. Proc AAAI-83109:113
4. Fukushima K, Miyake S (1982) Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern Recogn 15(6):455–469
5. Graves A (2012) Long short-term memory. Springer, Heidelberg
6. Heaton JB, Polson NG, Witte JH (2016) Deep learning in finance. arXiv preprint arXiv:160206561
7. Hebb DO (1949) In the organization of behavior, a neuropsychological theory. Wiley, New York
8. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
9. Hopfield JJ (1982a) Neural networks and physical systems with emergent collective computational abilities. Proc Nat Acad Sci U.S.A 79(8):2554–2558

10. Jordan MI (1997) Serial order: a parallel distributed processing approach. Adv Psychol 121:64
11. Kobayashi S, Shirayama S (2017) Time series forecasting with multiple deep learners: Selection from a bayesian network. J Data Anal Inform Process 05(3):115–130
12. Kohonen T (1981) Automatic formation of topological maps in a self-organizing system, pp 25–32
13. Minsky ML, Papert S (1969) Perceptrons: an introduction to computational geometry. The MIT Press
14. Nolfi S, Elman JL, Parisi D (1994) Learning and evolution in neural networks. Adapt Behav 4(1):5–28
15. Pitts W (1988) A logical calculus of the ideas immanent in nervous activity. MIT Press
16. Rosenblatt F (1958) Two theorems of statistical separability in the perceptron. In: Mechanisation of thought processes, Vol 1. HM Stationery Office, London
17. Rumelhart DE, Mcclelland JL (1986) Parallel distributed processing: explorations in the microstructure of cognition, vol 1 foundations. Language 63(4)
18. Von Neumann J (1951) The general and logical theory of automata. In: Papers of John Von Neumann on computing & computer theory, pp 1–41
19. Werbos P (1974) Beyond regression: new tools for prediction and analysis in the behavioral science. PhD dissertation Harvard University 29(18):65–78
20. Widrow B, Hoff ME (1988) Adaptive switching circuits in 1960 ire wescon convention record, 1960. Neurocomputing (Reprinted in)