Research in Economics and Management
ISSN 2470-4407 (Print) ISSN 2470-4393 (Online)
Vol. 4, No. 1, 2019
www.scholink.org/ojs/index.php/rem

Original Paper

Comparative Research on Influencing Factors of LSTM Deep Neural Network in Stock Market Time Series Prediction

TANG Yin[1], YANG Jinyu[1] & CHEN Jian[2*]

[1] Management School, Jinan University, Guangzhou, China

[2] Xinhua Collge, Sun Yat-sen University, Guangzhou, China

[*] CHEN Jian, Xinhua Collge, Sun Yat-sen University, Guangzhou, 510507, China

Received: January 4, 2019 Accepted: January 19, 2019 Online Published: January 24, 2019

doi:10.22158/rem.v4n1p84 URL: http://dx.doi.org/10.22158/rem.v4n1p84

Abstract

During training process of LSTM, the prediction accuracy is affected by a variation of factors, including the selection of training samples, the network structure, the optimization algorithm, and the stock market status. This paper tries to conduct a systematic research on several influencing factors of LSTM training in context of time series prediction. The experiment uses Shanghai and Shenzhen 300 constituent stocks from 2006 to 2017 as samples. The influencing factors of the study include indicator sampling, sample length, network structure, optimization method, and data of the bull and bear market, and this experiment compared the effects of PCA, dropout, and L2 regularization on predict accuracy and efficiency. Indice sampling, number of samples, network structure, optimization techniques, and PCA are found to be have their scope of application. Further, dropout and L2 regularization are found positive to improve the accuracy. The experiments cover most of the factors, however have to be compared by data overseas. This paper is of significance for feature and parameter selection in LSTM training process.

Keywords

stock market series prediction, LSTM deep neural network, parameter, PCA, optimization

1. Introduction

In recent years, there have been many studies with LSTM, some of which especially suitable for stock market time series forecasting (Hochreiter & Schmidhuber, 1997). The complex model structure and its huge number of parameters, however, have serious effect on the accuracy and performance. In practice, the selection of various factors such as training samples, model structure, and optimization methods is often subjective. As a result, it has become the biggest problem in engineering applications. Especially in multi-dimensional scenarios such as stock market time series forecasting, the influencing factors are more complicated.

The research on the influencing factors of LSTM model prediction accuracy, includes sample characteristics, network structure selection and optimization methods. The choice of influencing factors depends on fairly different application. Rao et al. use LSTM to classify texts on multiple social media platforms, comparing the effects of optimization methods, batch sizes, and activation functions on model performance (Rao & Spasojevic, 2016). Maknickienė et al. used LSTM in the USD/JPY exchange rate forecast (Maknickienė, Rutkauskas, & Maknickas, 2011). They found that neurons amounts and the number of training iterations were basically stable over a certain range. The hybrid model of ARMR and RNN built by Rather et al. uses the yield of 6 stocks as the model input, which can achieve higher precision than RNN (Rather, Agarwal, & Sastry, 2015). Xiong et al. used the yields and volatility of daily Standard & Poor's 500 index, and used 25 Google trends which reflect the trends of various major domestic industry and economy to predict the volatility of the Standard & Poor's 500 index, as well as found that Adam can achieve better accuracy (Xiong, Nichols, & Shen, 2016). To sum up, the above research is oriented to different fields, covering the training factors, network structure, neurons amounts, optimization methods and other influencing factors, but there are widespread problems such as insufficient training samples, insufficient factor selection, etc. In the stock market time series forecasting, the sample and indicators selection in different time intervals may get completely different prediction conclusions, so it is necessary to make more complete verification about the above factors.

Generally, reducing network errors by increasing the number of hidden layers, but it is prone to "over-fitting" and complicates the network. It is often subjective that selection of hidden layers amount and neurons quantity. Angel made an attempt in this aspect, and proposed an estimation method for the number of hidden layer nodes by using interval estimation (Angel, 2017). Thomas proposed a self-organizing cognitron to identify an optimal set of neurons in a hidden layer and optimal number of hidden layers in neural model using gradient decent based on number of neurons and error at each iterations (Thomas, Manoj, & Annappa, 2016). The mathematical model proposed by Wagarachchi can dynamically drop the hidden layer during training (Wagarachchi & Karunananda, 2016). This paper will refer to this conclusion for selecting hidden layers amount.

Recent research have shown that some adaptive learning rate optimization methods are not superior to SGD (Stochastic Gradient Descent), experiments on ResNet and other networks by Keskar et al. show that the Adam, Adagrad or RMSprop methods only work well in the initial stage, and they propose a method of dynamically switching to SGD (Keskar & Socher, 2017). Feng compared SGD, MSGD, AdaDdelta, AdaGrad, Adam, RMSprop and other methods in the question answer field experiment, and received the similar results (Feng, Xiang, & Zhou, 2015). But in LSTM experiment, Adam is better (Andrychowicz et al., 2016). Recently proposed techniques such as Dropout (Srivastava et al., 2014; Bluche, Kermorvant, & Louradour, 2015) and regularization (L2 regularization) (Theodoridis, 2015) have effectively alleviated the problem of DNN training overfitting. This paper will further compare the optimization methods.

This paper takes the 2006-2017 stock data of Shanghai and Shenzhen 300 constituents as a sample, and built a model based on LSTM deep neural network. This paper made the systematic comparison of several factors affecting the prediction accuracy. And it test the influence of principal component analysis and optimization strategies such as dropout and L2 regularization on prediction accuracy.

2. Research Ideas and Framework

At present, there is no systematic study on the influence of specific input indicators selection, model structure and optimization method on accuracy in model training. In this paper, the LSTM deep neural network is used to predict the stock market time series, and the factors affecting accuracy are studied through multiple sets of contrast experiments. The optimal training samples, model structure and optimization methods are selected to improve the prediction accuracy. Selection of indicators further consider the differences between the bull and bear markets.

The overall thought is to conduct multiple contrast experiments on the basis of baseline experiments, which is the study on impact of model accuracy by researching training samples, network structure, or optimization .

a. Comparison of indicators: For input characteristics, the transaction basic data, technical indicator data, transaction basic data + market data, technical indicator data + market data, transaction data + market data + technical indicators, and all indicators as input and comparison the rate of accuracy to select the best input characteristics (baseline experiment);

b. The PCA was used to reduce the dimensionality of 28 indicators, and the processing results were used as input variables to compare with the corresponding non-dimensionality reduction models;

c. Change sample length for examining the effect of length on prediction accuracy;

d. Change training samples quantity for examining the effect of sample size on prediction accuracy;

e. Model structure comparison: After the previous step, choseing the most optimal model, and adjusting the number of neurons in the hidden layer to research the influence of different hidden layers neurons amount on the prediction accuracy;

f. Comparison of optimization methods: The SGD, RMSprop and Adam methods are used to optimize the network training process, and to study the influence of different optimization methods on prediction accuracy;

g. Comparison of the bull and bear market: Compare the effects of models in different market status, and optimize the model with dropout and L2 regularization techniques.

2.1 Variable Selection

Usually, the market broadly divides the indicators into: (1) Various basic data related to stock trading; (2) Statistical technical indicators derived from transaction data; (3) An index related to the macro situation of the stock market; (4) Fundamental factors. Indicator (1) to (3) are collectively called as trading indicators, and (4) is financial indicators. The selection principle of trading indicators must include complete basic data, and select some representative indicators in trend indicators and emotional indicators. The Shanghai and Shenzhen 300 Index represents the comprehensive situation of the two domestic markets, and also includes the common financial indicators in the market. The details are shown in Table 1.

Table 1. Trading Indicator

Type of Indicators	Indicators	Abbreviation	Calculation formula
Ⅰ. Basic Trading Indicators	Opening price of the day	Open	
	Highest price of the day	High	
	Lowest price of the day	Low	
	Closing price of the day	Close	
	Quote change	Change	
	Volume	Vol	
Ⅱ. Technical Indicators	Turnover Rate	TR	TR = Vol/Total shares in circulation * 100%
	Simple Moving Average	SMA	Moving average within N day=The sum of the closing prices on the N day/N
	Moving Average Convergence Divergence	MACD	EMA(12) = EMA of Previous day(12) × 11/13 + Close × 2/13
		DIF	DIF = EMA(12)- EMA(26)
		DEA	DEA = DEA of Previous day × 8/10 + DIF of Present day × 2/10
	Stochastic Oscillator	KDJ_K	K value of Present day = 2/3×K value of Previous day + 1/3×RSV of Present day
		KDJ_D	D value of Present day = 2/3× D value of Previous day + 1/3×K value of Present day
		KDJ_J	J value of Present day = 3×K value of Present day - 2×D value of Present day
	Bias Ratio	BIAS	BIAS =(Close－Moving average closing price within N day)/Moving average closing price within N day×100％(3-12)6RSI
	Relative Strength Index	RSI	RSI = The sum of the closing gains in N days/(The sum of the closing gains in N days + The sum of the closing declines in N days) ×100%
	Rate of change	ROC	ROC = Close/N days ago closing price

	Psychological line	PSY	PSY = Rising days within N day/N×100
Ⅲ. Macro Index (CSI300 Index)	Opening price of the day	Open_300	
	Highest price of the day	High_300	
	Lowest price of the day	Low_300	
	Closing price of the day	Close_300	
	Quote change	Change_300	
	Volume	Volume_300	

**Note*. Stochastic N day RSV = (Closing price within N day-Lowest price within N day)/(High price within N day-Lowest price within N day) * 100%.

The financial indicators adopt several indicators of Price to Earning Ratio, Price to Sales Ratio, Price Cash Flow Ratio and Price to book Ratio, the details are shown in Table 2.

Table 2. Financial Indicator

Type of indicators	Indicators	Abbreviation	Calculation formula
Ⅳ financial indicators	Price to Earning Ratio	PE	PE = Share price/Earnings per share attributable to the parent company in the last 12 months
	Price to Sales Ratio	PS	PS = Share price/Operating income per share for the last 12 months
	Price Cash Flow Ratio	PC	PC = Share price/Operating cash flow per share for the last 12 months
	Price to Book Ratio	PB	PB = Share price/Recent earnings per share net assets

2.2 Distinguish the Bull and Bear Market

Considering that market reactions may be inconsistent under different market conditions, this paper draws on the nonparametric method of He Xingqiang et al. (He & Zhou, 2006) to find the crests and troughs of stock market index changes. Set the monthly average of the stock market index as p_t^m.

Definition: When if and only if p_t^m is the maximum value in a time window with a width of 3 months, p_t^m is crest; the same, when if and only if p_t^m is the minimum value in a time window with a width of 3 months, p_t^m is trough.

The crests and troughs need to alternate, so the lower price of the connected crests and the higher price in the connected troughs are eliminated. In addition, in order to eliminate the false bull and bear market cycles in the future, we will not miss the big bull and big bears that correspond to the big rise and fall in the short term. The bull and bear markets will follow the requirements below: *(1) Except for bull or bear market cycles with a period of less than 6* months*; (2) Except for crests and troughs that are less* *than 4 months from the endpoint; (3) If the one-way duration of a bull or bear market does not exceed 4 months, the price change before and after the price reversal must be greater than 20%.*

3. Experimental Results

3.1 Model Structure

(1) Input layer and output layer selection: This article uses the daily data within the previous N days to predict the average increase and decrease in the next 3 days. There are 28 input variable form 4 categories, and X is selected after screening (X <= 28); Output layer has three neurons (2 represent large rises, 1 represent small rises and falls, and large falls by 0).

(2) Hidden layer selection: Considering the length of the time series itself, this paper set two hidden layers. When the number of two hidden layer nodes is similar in the double hidden layer network, the network training effect is best. Set the initial hidden layer node number to 136. Based on this, the comparison experiment adjusts the number of hidden layer nodes to observe influence of model structure on model accuracy.

(3) Hyperparameters: The time series length is set to 30. Select RMSprop as the optimization method and use Categorical Cross-Entropy (Boer et al., 2005) as the loss function, batch size is set to 32, the number of iterations is set to 30. The model is a multi-classification model, and the model accuracy evaluation uses the default accuracy. Accuracy measures the correct proportion of the classification and represents the performance of model. Let $\hat{y}_i$ be the prediction category of the i_{th} sample, and y_i is the real category of the i_{th} sample. Defined the accuracy rate on n samples:

$$accuracy = \frac{1}{n}\sum_{i=1}^{n} 1(\hat{y}_i = y_i) \tag{1}$$

Where 1(x) means that the value is 1 when the prediction result is consistent with the real result, otherwise, 1(x) is 0.

3.2 Data Set Selection and Preprocessing

The original data was collected from the CSMAR database. There is large differences in China's stock market system before and after 2005, this paper selects the Shanghai and Shenzhen 300 Index and its constituent stocks from January 1, 2006 to January 19, 2017.

(1) Preliminary screening: Removed 10 stocks with missing financial indicators in 2005 and retained 290 stocks;

(2) Calculate technical indicators and forecast targets, which is rise and fall of the average closing price of stocks in the next three days;

(3) Need to merge the CSI 300 Index based on date and individual stock data;

(4) The Max-Min method is used to perform dimensionless processing on 28 types of feature data, and the data is segmented into standard input data with a sequence length of 30, and finally 592756 samples are obtained.

(5) Extract the column Change where the price is up and down, and calculate the average rise and fall extent of each stock in the next three days. The average price of the next three days on the t*th* day set as $\overline{c}_{t3}$. Then, in order to make the number of each type of training samples similar, take the front and back tertiles of all stock ups and downs (AC), which is recorded as 0.33AC, 0.67AC.

·If $\overline{c}_{t3}$ < 0.33AC, marking the sample as 0;

·If 0.33AC <= $\overline{c}_{t3}$ < 0.67AC, marking the sample as 1;

·If $\overline{c}_{t3}$ >= 0.67AC, marking the sample as 2.

(6) The data is randomly scrambled, and then 80% of them are taken as training data. There is 3/4 of them are further used as train_data (355653), thereby training the model; take 1/4 of them as the verification data val_data (118551) to compare contrast model. The remaining 20% of data as the final test data (118552) to test the accuracy and stability of the model. The optimization process is Rmsprop.

3.3 Implementation Process

This experimental uses NVIDIA CUDA programming techniques to accelerate the training of deep neural networks. The experimental environment of this paper is CPU: Intel i7-6900K, GPU: TITAN X, memory: 4*12G, operating system is Ubuntu, data processing and model algorithm are written in python, using keras based on tensorflow framework.

4. Result

4.1 Comparison of Indicators (Baseline Experiment)

According to the above four types of indicators, setting the following six comparison models to select the best input features.

Table 3. The Influence of Input Characteristics on Model Accuracy (Compared to Random Model Accuracy = 33.33%)

Model	dimension	Indicator selection	accuracy
M1	6	Ⅰ Basic Trading Indicators	51.94%
M2	12	Ⅱ Technical Indicators	44.36%
M13	12	Ⅰ Basic Trading Indicators Ⅲ Macro Index（CSI300 Index）	62.40%
M23	18	Ⅱ Technical Indicators Ⅲ Macro Index（CSI300 Index）	60.58%
M123	24	Ⅰ Basic Trading Indicators Ⅱ Technical Indicators Ⅲ Macro Index（CSI300 Index）	61.00%
M1234	28	Ⅰ Basic Trading Indicators Ⅱ Technical Indicators Ⅲ Macro Index（CSI300 Index） Ⅳ financial indicators	60.70%

**Note*. This table select the highest value in the 30 iterations of the model, the same below.

As Table 3 shown, the accuracy of M1 is 19% higher than that of the stochastic model, indicating that it is effective to predict the rise and fall of stock based on the basic transaction indicators. The M2 accuracy rate is lower than M1, probably because the technical indicators have lost information on the original trading indicators. Moreover, the accuracy of the M2 model is unstable compared to M1, which may be related to the inclusion of more information noise and the 10% rise and fall limit of China's stock market (shown as Figure 1 and Figure 2). Considering that the training samples of both models are 470,000, while M1 is 6 dimensions and M2 is 12 dimensions, the slight overfitting of M1 is related to the too small dimension of input data.

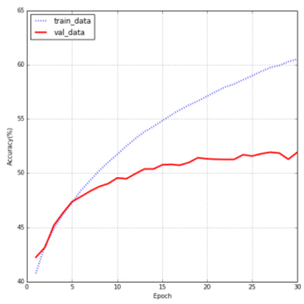


Figure 1. Result of M1

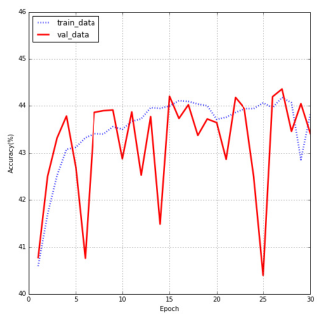


Figure 2. Result of M2

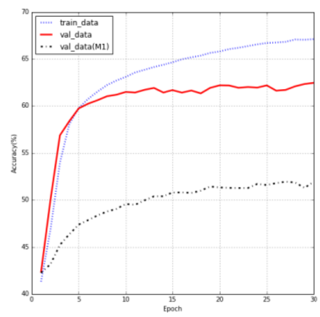


Figure 3. Result of M13

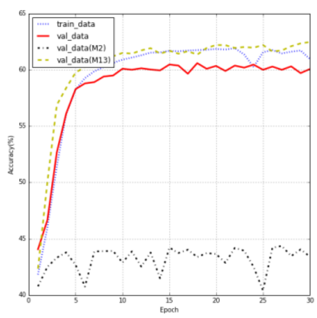


Figure 4. Result of M23

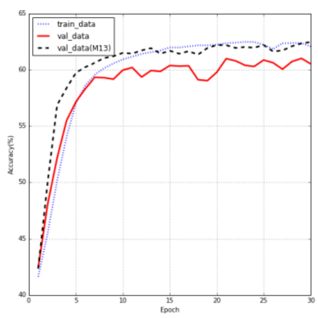


Figure 5. Result of M123

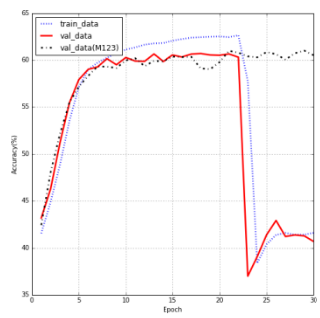


Figure 6. Result of M1234

*In the picture, Epoch on the horizontal axis represents the number of iterations, and the vertical axis Accuracy represents the accuracy of the model (accuracy), the same below.

The accuracy of M13 is 11% higher than that of M1 (Figure 3) and the accuracy of M23 is 16% higher than that of M2 (Figure 4), indicating that the index has a great influence on the stock price trend of individual stocks. The accuracy of M123 is lower than that of M13 (Figure 5), and the input information redundancy may cause model accuracy to decrease. The accuracy of the validation set of M1234 is slightly lower than that of M123 (Figure 6), indicating that the company's financial indicators have no reference value for short-term forecasts of stock prices. In addition, with the increase of iteration, M1234 with a large input data dimension accuracy has a rapid decline both training set and verification set. It is easy to appear gradient explosion when model has large input data dimension and redundant input data information.

4.2 PCA before and after Dimensionality Reduction

Redundant and high dimension of input information may reduce the accuracy of deep neural network and affect its predictive ability. Table 4 compares the model training after performing principal component analysis dimension reduction (retaining 99% of the information).

Table 4. Principal Component Analysis before and after Dimensionality Reduction

Post-dimensionality Model	Dimension	Accuracy	Original Model	Original Dimension	Accuracy Improvement
M1234-PCA	17	62.01%	M1234	28	+1.31%
M23-PCA	12	61.56%	M23	18	-0.98%
M13-PCA	6	60.96%	M13	12	+1.5%

As seen from Figure 7 to Figure 9, the overall dimensionality reduction can improve the accuracy. However, the accuracy of the model M13-PCA is lower than that before the dimension reduction (model M13), indicating that the above conclusions only apply to the case of more dimensionality. Additionally, compared with the model before dimension reduction, the model shows over-fitting after dimensionality reduction, indicating that the lower the input dimension, the more likely the over-fitting will occur.

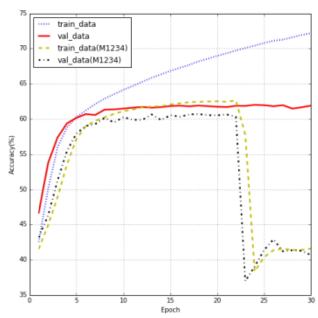


Figure 7. Result of M1234-PCA

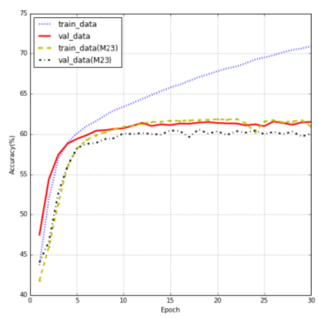


Figure 8. Result of M23-PCA

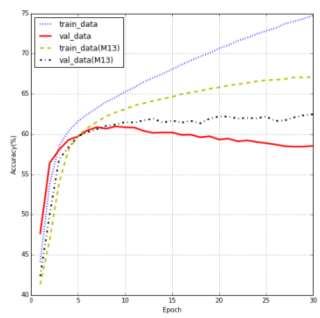


Figure 9. Result of M13-PCA

4.3 Sample Length Comparison

Based on the above results, the data is processed into different time series length such as 10 (M13-10), 20 (M13-20), 40 (M13-40), 50 (M13-50) for the M13 model, and training model (Table 5). Comparing with Table 3, the accuracy of M13-10 and M13-20 is slightly decreased, while small difference between M13-40 and M13-50. As shown in Figure 10 to Figure 13, the learning rate of M13-10 is slower, while the learning speed or over-fitting of M13-20, M13-30, M13-40, and M13-50 are basically consistent, but M13-40 appears gradient explosion. The phenomenon of gradient explosion, which also shows that the length of the sequence is too long will increase the possibility of gradient explosion.

Table 5. The Effect of Sequence Length on Model Accuracy (Compared to the Accuracy of M13 Is 62.46%)

Model	Series Length	Accuracy
M13-d10	10	60.73%
M13-d20	20	61.65%
M13-d40	40	62.19%
M13-d50	50	62.73%

After the 20th iteration, the model has converged, and the M13-40 has a gradient explosion after the 24th iteration. As shown in Figure 14, comparing the 20th training result of four models, as the length of the sequence increases, the accuracy of the verification set increases first and then remains unchanged. This means that the sequence length is too short, and the hidden information in training sample may not be learned. The longer sequence length, the more noise data may be contained, which may result in reduced prediction accuracy.

Further, as can be seen from Figure 10, as the number of iterations increases, the over-fitting phenomenon of the model M13-d10 becomes more apparent, not only is the input data dimension too small, and the short time series length can also lead to overfitting. In the models M13 and M13-d50 with the highest accuracy, since the training time of M13-d50 is 16 hours, the training time of M13-d30 is 10 hours, and the accuracy of the two is similar. Considering the time cost, the following model training still sets the sequence length to 30.

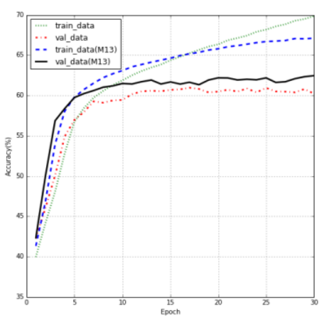


Figure 10. Result of M13-d10

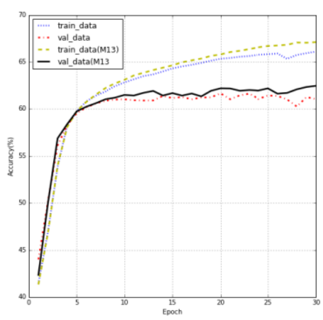


Figure 11. Result of M13-d20

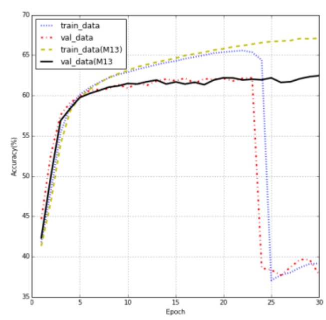


Figure 12. Result of M13-d40

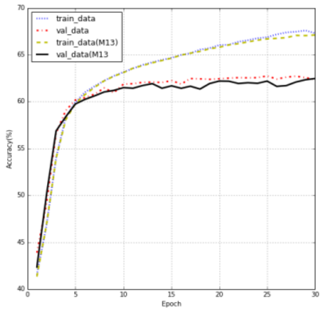


Figure 13. Result of M13-d50

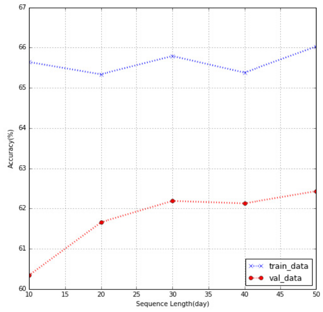


Figure 14. Effect of Sequence Length on Model Accuracy

4.4 Sample Size Comparison

In general, the more training samples, the better the accuracy of the model, but the noise contained in the training data will also increase. In this paper, 60,000 (M13-6), 180,000 (M13-18), 300,000 (M13-30), and 590,000 (M13-59) samples were randomly selected for training (Table 6), respectively, and extracting 25% of them as a verification sample. Comparing with the accuracy of M13 (62.46%, 470,000 samples), it can be seen that the sample size does have a certain impact on the accuracy of model.

Table 6. The Effect of Sample Size on Model Accuracy (Compared to M13 Accuracy of 62.46%, 470,000 Samples)

Serial number	Model	Sample Amount	Accuracy	Comparing with M13
1	M13-6	60,000	52.56%	-9.90%
2	M13-18	180,000	59.66%	-2.80%
3	M13-30	300,000	60.97%	-1.49%
4	M13-59	590,000	62.45%	-0.01%

As shown in Figure 15 to Figure 18, the models M13-6, M13-18, M13-30, and M13-59 start to converge at the 25th, 12th, 8th, and 5th epoch, respectively, M13 starts convergence at the 5th iteration, implying that the quantity of training samples is positively correlated with the convergence speed. The over-fitting phenomenon of M13-18 is more serious than that of M13, indicating that the less the training sample, the more likely it is to overfit. In addition, after the 20th iteration, the accuracy of validation set has converged. Comparing the 20th training results of the five models (Figure 19), as the number of samples increases, the accuracy of the verification set increases first and then remains unchanged. When the sample is large enough, continuing to increase the number of samples doesn’t improve model accuracy model.

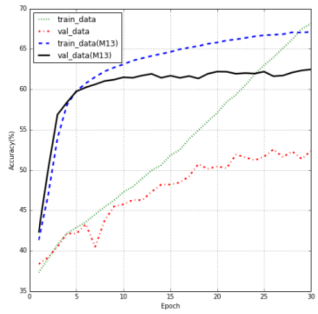


Figure 15. Result of M13-6

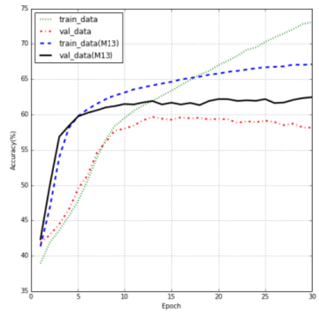


Figure 16. Result of M13-18

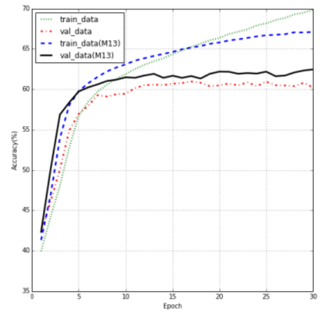


Figure 17. Result of M13-30

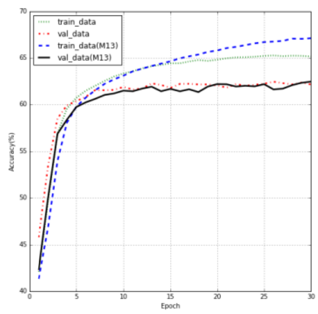


Figure 18. Result of M13-59

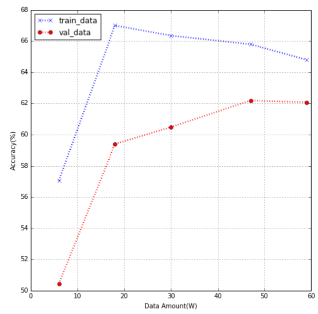


Figure 19. The effect of sample size on model accuracy

4.5 Network Structure Comparison

Compare and analyze the models with different numbers of hidden layer neurons and select the optimal parameters. The number of hidden layer neurons was set to 16 (M13-h16), 56 (M13-h56), 96 (M13-h96), 176 (M13-h176), and 216 (M13-h216). As shown in Table 7, reducing the number of hidden layer neurons declines accuracy, while increasing the number of hidden layer neurons , accuracy rate of the model is not obvious, it and does not improve.

Table 7. The Influence of the Model Structure on the Accuracy of the Model (Compared with M13 Accuracy of 62.46%, 136 Nodes Per Hidden Layer)

Model	Model Structure	Accuracy
M13-h16	12-16-16-3	54.54%
M13-h56	12-56-56-3	60.29%
M13-h96	12-96-96-3	60.86%
M13-h176	12-176-176-3	61.27%
M13-h216	12-216-216-3	61.42%

As shown in Figure 25, comparing the 20th training results of the five models, it can be seen that excessive or too small number of hidden layer neurons will reduce the accuracy of the model. In addition, as can be seen from Figure 20 to Figure 24, the more the number of neurons, the faster model convergence speed.

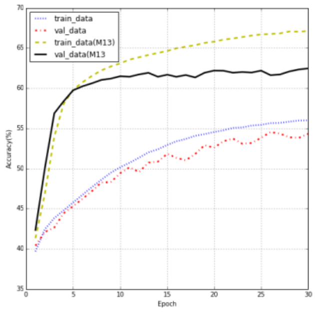


Figure 20. Result of M13-h16

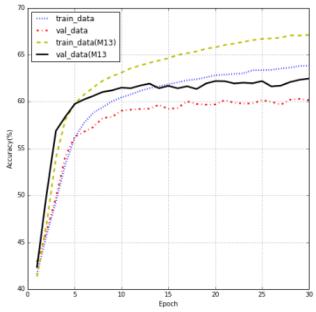


Figure 21. Result of M13-h56

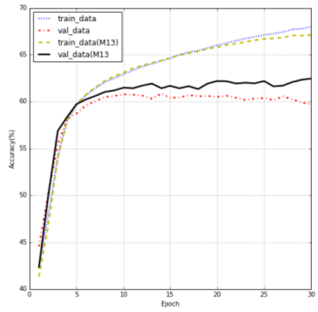


Figure 22. Result of M13-h96

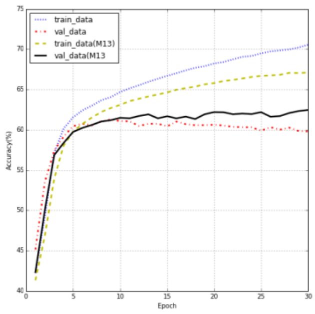


Figure 23. Result of M13-h176

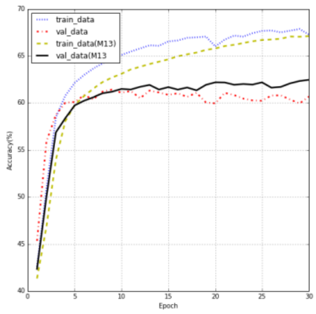


Figure 24. Result of M13-h216

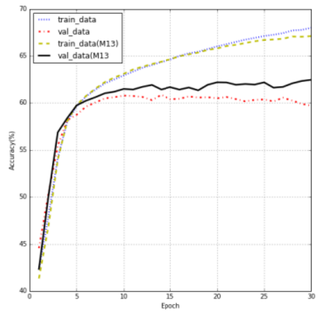


Figure 25. The influence of model structure on accuracy

4.6 Optimization Method Comparison

SMD (M13-SGD) and Adam (M13-Adam) with the Rmsprop are compared as benchmark experiment. As shown in Table 8, the accuracy of M13-SGD is slightly lower than that of the benchmark experiment M13, and accuracy of M13-Adam is not significantly different from that of M13. As shown in Figure 26, M13-SGD converges from the 20th epoch, and M13 converges from the 5th iteration, indicating that Rmsprop converges faster than SGD. As shown in Figure 27, using Adam and Rmsprop, the model has little difference in convergence speed and prediction accuracy. However, when Adam is used, the over-fitting phenomenon tends to occur as the number of iterations increases. Therefore, when the sample size is small, Rmsprop is more suitable.

Table 8. The Effect of the Optimization Method on the Accuracy of the Model (Compared to the Benchmark Experiment M13 Accuracy Rate 62.46%, Rmsprop Method)

Model	Optimization	Accuracy
M13-SGD	SGD	61.40%
M13-Adam	Adam	62.08%

After comparing the training samples, model structure and optimization methods, the optimal model M13 is selected according to the accuracy of the verification set, finally, using the test set data verify the effect of the model. After verification, the accuracy of the test set is 62.31%, which is only slightly lower than the accuracy of the verification set of 62.46%, which further proves the stability of the optimal model.

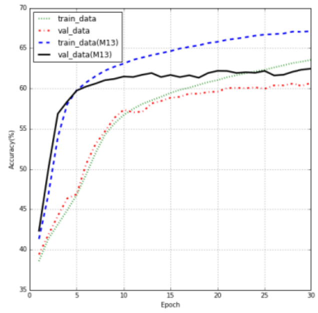


Figure 26. Result of M13-SGD

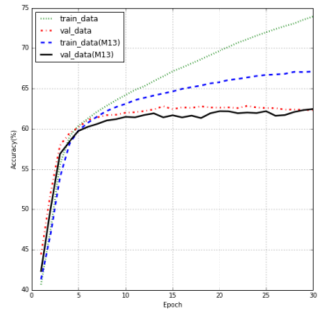


Figure 27. Result of M13-Adam

4.7 Comparison of Bull and Bear Market and Its Optimization

Using the bull-bear market discriminant method to discriminate between the CSI 300 index from January 2006 to January 2017. The interval of bull and bear market is as follows:

Table 9. Market Status Division Result

Type of Market	Interval of Time
	2006.1-2007.10
	2008.12-2009.11
Bull	2010.8-2011.4
	2012.12-2013.2
	2014.4-2015.6
	2007.11-2008.11
	2009.11-2010.7
Bear	2011.5-2012.11
	2013.3-2014.3
	2015.7-2017.1

The total sample size of the bull market was 215,591, and that of the bear market was 319,118. At the same time, in order to remove the influence of the sample size on the model accuracy, 215,591 samples (M13-22) and 319118 samples (M13-32) were randomly selected from 592756 samples in the mixed market status for model training. Since the number of training samples in each market status is small after dividing the market status, 65% of the data is used as the training sample (train_data) and 35% of the sample (val_data) is used as the verification sample.

Table 10. Different Market Status Model Results (Accuracy Rate of Benchmarking Experiment M13 Is 62.46%)

Model	Market Status	Accuracy	Contrast Model	Market Status	Accuracy	Sample Amount
M13-n	bull	61.90%	M13-22	mixed	59.45%	215591
M13-x	bear	64.14%	M13-32	mixed	60.49%	319118
M13-n(dropout)	bull	62.24%				
M13-n(regularization)	bull	61.54%				

As shown in Table 10, the accuracy of M13-n with bull market data as input is slightly increased by 2.5 percentage points over the mixed market statu model M13-22 with the same sample size. As shown in Figure 28, the training speed of the bull market model is slightly faster than that of the mixed status. The accuracy of the bear market as the input model M13-x is slightly higher than the accuracy of the mixed market status model M13-32 of the same sample size by 3.5 percentage points, indicating the behavior of Chinese investors may be more consistent in bear market. As shown in Figure 29, the convergence rate of the bear market model is significantly faster than that in the mixed status. It can be inferred that increasing the number of bear market samples can further improve the prediction accuracy of the model. Combining Figure 28 and Figure 29, it is explained that subdividing the data by market status can improve the model accuracy and improve the training speed of the model.

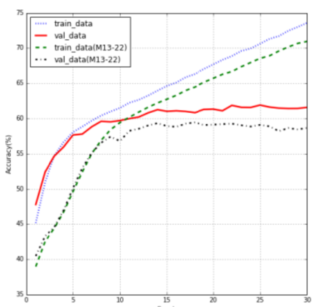


Figure 28. Result of M13-n

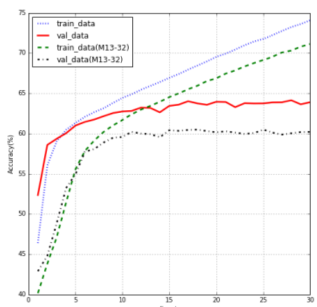
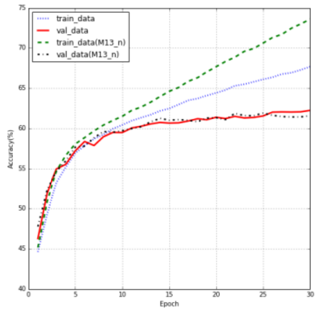


Figure 29. Result of M13-x



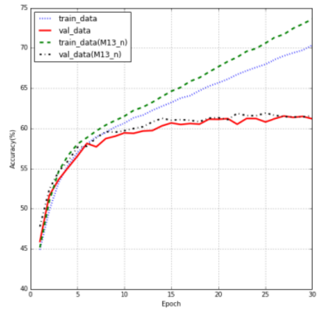


Figure 30. Result of M13-n (dropout) **Figure 31. Result of M13-n (regularization)**

In addition, due to the classification, the number of samples in the two categories is greatly reduced, and as the number of iterations increases, the model appears to be over-fitting. Therefore, M13_n is optimized by dropout and L2 regularization techniques (M13_n (dropout) and M13_n (regularization)). M13_n (dropout) has a slightly higher accuracy than M13-n, while M13_n (regularization) has a slightly lower accuracy than M13-n. Comparing Figure 30 with Figure 31, it is found that both methods alleviate the overfitting, but dropout works better. Therefore, for the model with large difference between the training set and the verification set, the dropout and L2 regularization techniques can alleviate over-fitting phenomenon of the model, but the accuracy of the model is not improved.

5. Conclusion

In this paper, we use the LSTM deep neural network to predict stock market time series, and the experiments to conduct a comparatively complete comparison of several factors affecting the prediction accuracy. In the LSTM deep neural network applied to stock market time series forecasting, indicators selection, PCA dimension reduction, sequence length, sample size, model structure and market cycle, which have a substantial impact on the prediction effect, and have a certain scope of application. This paper gives specific recommendations on the above influencing factors.

The contribution of this paper is to systematically study various factors affecting the prediction accuracy of LSTM deep neural network from four aspects: indicators selection, training sample, network structure and optimization method. It provides comprehensive reference for the training of LSTM and parameter adjustment of stock market time series prediction. The paper is only tested on the Shanghai and Shenzhen 300 constituent stocks, and the more efficient conclusions will depend on more complete data experiments at domestic and abroad. Furthermore, the next step in the research will be to compare the back-propagation strategy with the evolution strategy in the training process.

References

Andrychowicz, M. et al. (2016). *Learning to Learn by Gradient Descent by Gradient Descent*.

Angel, K. M. (2017). Closed Determination of the Number of Neurons in the Hidden Layer of a Multi-Layered Perceptron Network. *Soft Computing*, *21*, 1-13.

Bluche, T., Kermorvant, C., & Louradour, J. (2015). Where to Apply Dropout in Recurrent Neural Networks for Handwriting Recognition? International Conference on Document Analysis and Recognition. *IEEE*, *2015*, 681-685. https://doi.org/10.1109/ICDAR.2015.7333848

Boer, P. T. D. et al. (2005). A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, *134*(1), 19-67. https://doi.org/10.1007/s10479-005-5724-z

Feng, M., Xiang, B., & Zhou, B. (2015). Distributed Deep Learning for Question Answering. *Computer Science*, *2015*, 2413-2416.

He, X. Q., & Zhou, K. G. (2006). Bull-Bear Market Period and Periodic Coordination between Stock Markets. *Management World*, *2006*(4), 35-40.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Keskar, N. S., & Socher, R. (2017). *Improving Generalization Performance by Switching from Adam to SGD*.

Maknickienė, N., Rutkauskas, A. V., & Maknickas, A. (2011). Investigation of Financial Market Prediction by Recurrent Neural Network. *Innovative Technologies for Science, Business and Education*, *2*(11), 3-8.

Rao, A., & Spasojevic, N. (2016). *Actionable and Political Text Classification using Word Embeddings and LSTM*.

Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent Neural Network and a Hybrid Model for Prediction of Stock Returns. *Expert Systems with Applications An International Journal*, *42*(6), 3234-3241. https://doi.org/10.1016/j.eswa.2014.12.003

Srivastava, N. et al. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.

Theodoridis, S. (2015). Chapter 18—Neural Networks and Deep Learning. In *Machine Learning* (pp. 875-936). Elsevier Ltd. https://doi.org/10.1016/B978-0-12-801522-3.00018-5

Thomas, L., Manoj, K. M. V., & Annappa, B. (2016). Discovery of Optimal Neurons and Hidden Layers in Feed-Forward Neural Network[C]//IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies. *IEEE*, *2016*, 286-291.

Wagarachchi, N. M., & Karunananda, A. S. (2016). Mathematical Modeling of Hidden Layer Architecture in Artificial Neural Networks. *International Proceedings of Computer Science & Information Tech*.

Xiong, R., Nichols, E. P., & Shen, Y. (2016). *Deep Learning Stock Volatility with Google Domestic Trends*.