



A monetary policy prediction model based on deep learning

Minrong Lu¹

Received: 30 March 2019 / Accepted: 18 June 2019 / Published online: 27 June 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

Applying neural network and error t-value test, this study trains and analyzes 28 interest rate changes of China's macro-monetary policy and the mutual influences between reserve adjustments and financial markets for 51 times from 2000 to 2018 according to the data correlation between financial market and monetary policy. Through the principal component analysis, the bilateral financial risk system and data set are established, and the data set pre-process and dimensionality reduction are carried out to extract the most informative features. Six training cases are designed with processed features, and then the cases are input to each neural network model for combined prediction. Firstly, based on backpropagation neural network (BP), the forecasting model of monetary policy is established. Then, considering the importance characteristics of financial index data, expert weights based on BP, are introduced to propose weights backpropagation (WBP) model. On the basis of the timing characteristics of financial market, the WBP model is improved and the timing weights backpropagation (TWBP) model is proposed. Experiments show that different training cases bring out various effects. The accuracy rate of interest rate and reserve change value is lower than the original value after training. The mutation after data processing affects the learning of neural network. At the same time, the WBP and TWBP models improve according to the importance and timing characteristics of financial indicators have less errors in results, and the TWBP model has higher accuracy. When the number of hidden layers is 3, good results can be obtained, but in manifold training of the timing cycle, the efficiency of that is not as good as the WBP model.

Keywords Neural network · Time series · Monetary policy · Financial risks

1 Introduction

The world financial market has experienced a series of turmoil, and brought out serious economic crises from the beginning when monopoly of funds dominated the financial market, to the period when political and economic development seeped into financial markets to the period of the globalization of financial markets [1, 2]. There are many reasons for the economic crisis and financial risk events. They have different omens in different historical stages. Among them, during the process of financial market globalization, the monetary policy of each country has the closest relationship with the financial market crisis [3–5].

This view has been accepted by most experts and scholars. Especially in terms of the economic bubble and the macro-control during the crisis period, there is a strong correlation between monetary policy and financial market [6, 3, 7]. Therefore, the prediction research of monetary policy is very necessary. It can provide early warning and prevention of financial risks.

In the existing research on monetary policy and financial market, scholars focus on the effectiveness of monetary policy, transmission mechanism and regulatory expectations [4]. Some scholars put forward the interest rate and money supply of monetary policy tools based on autoregressive integrated moving average model (ARIMA) in the application of models and methods, and the analysis of monetary policy adjustment expectations and fund supply and demand in the economic cycle based on Keynes economic cycle theory [8–11]. Some scholars have also established new ones. Methods and models, such as the

✉ Minrong Lu
1698030620181@fjxxu.edu.cn

¹ School of Accounting, Fujian Jiangxia University, Fuzhou 350108, Fujian Province, China

proposed inflation prediction target method, generalized method of moments (GMM), dynamic stochastic general equilibrium model (DSGE), etc., all play a good role in the regulation of monetary policy [12–14]. Some scholars have proposed using the ARIMA model to decompose the interest rate of monetary policy tools and money supply, and using the Keynes economic cycle theory to analyze the monetary policy regulation expectations and the supply and demand of funds in the economic cycle [15]. Some scholars have also established new methods and models, such as inflation prediction target method, GMM model, and DSGE model [16, 12, 13], which all play a good role in guiding monetary policy regulation. In recent years, with the rapid development of information technology, the combination of computer-related tools and financial markets has become more and more close. The emergence of big data, artificial intelligence and blockchain technology has changed the traditional trading system of financial markets, for instance [17, 18]. Research in the field of neural networks and finance has mainly focused on stock market, exchange rate, house price, evaluation of monetary policy transmission effectiveness based on BP neural network (networks prototype), and expansionary monetary policy of USA based on long short-term memory (LSTM) neural network research, China and the USA to cope with the financial crisis of the monetary policy (using the IS-LM-BP model). There are many researches on the improvement of neural networks in the non-financial field. In the financial field, research has focused on the combination of time series and improved prediction of financial markets by convolutional neural networks [19–21]. Existing research in the field of neural networks and finance favors empirical research on economic theory, focusing on the comparative analysis of the effects of model applications or parameter adjustment in neural networks, and lacks adaptability to neural networks. At the same time, neural networks have little research on monetary policy prediction, including the processing of financial data in time shift analysis, and neural network algorithm improvement.

This paper mainly combines big data technology to study the bilateral risk of financial market and use neural network to predict monetary policy and provides suggestions for prevention and control of financial risk, in order to train better analytical forecasting model to serve financial market on the basis of neural network. Main contribution of this paper includes: (1) established bilateral risk indicators of financial markets based on financial expert system, collate relevant data sets, and at the same time pre-processed; (2) applied BP to monetary policy forecasting; (3) set external weights to improve BP based on financial expert system, and further proposed WBP model; (4) further improved WBP according to the principle of financial market time validity, and proposed TWBP model.

The structure of this paper is as follows: Sect. 3 introduces the construction of the bilateral financial risk system based on principal component analysis (PCA), and the extraction of hidden risk elements in the data through data mining as well as the establishment of data sets. In Sect. 4, the author will apply BP to monetary policy forecasting and combine characteristics of life cycle of financial market and related weight to improve model. The test results and effect analysis of these models will be explained in Sect. 5. According to the model training results, in Sect. 5 will put forward the recommendations and measures for financial risk prevention, so as to realize the value of prevention and control.

2 Related work

In recent years, applications of big data have been very extensive. Well-known high-tech companies such as Microsoft, IBM, Oracle, Google, and Baidu are constantly studying and exploring this subject [22, 23]. The development of computer technology is gradually pushing the global financial market to a new era. The big data and neural network are increasingly mature in research and application in terms of financial market and financial risks, and have been adopted by more and more government departments and financial institutions [24, 25, 18]. In 2012, a plan for global big data began to emerge. The combination of data mining and traditional business models has had a huge impact on financial markets. The relatively rapid development of machine learning technology has been favored by financial researches ever since.

After the rise of big data, technologies such as support vector machine (SVM), multilayer perceptron (MLP), and deep learning (DL) have been continuously renovated. Some studies have shown that these techniques improved the classification effect and changed the traditional prediction [19, 26, 27, 28]. Deep learning, especially multilayer neural network-based learning methods, is more outstanding in improving the accuracy of classification or prediction when solving complex problems [29–31]. Since the initiate of BP neural network, many scholars have attempted to expand the research of neural network theory into the financial field and carried out various application research from the perspective of neural network technology: static neural network (SNN) [32] was used for weekly data forecasting of exchange rates; dynamic neural network (DNN) [33] was used to analyze financial data to improve forecasting ability; recurrent neural network (RNN) was applied to financial news event text information processing; long short-term memory neural network (LNN) was applied to long-term timing dynamic information processing; convolutional neural networks (CNN) were introduced

into the futures market; deep belief network (DBN) used the oscillator box theory to build an automatic stock decision support system; and so on [34, 33, 35, 36, 37, 32].

Deep learning is a class of algorithms that processes data like the way the human brain does [38, 39]. Some studies have found that multilayer training through neural networks which empirically combines relevant models can effectively improve financial market forecasting and risk prevention capabilities. Domestic and foreign scholars have launched a series of studies of the application in the financial market: in 2013, Asness et al. applied deep learning to a variety of products such as exchange rates, commodities, and bonds. In references [34, 35], the authors demonstrated the applicability of deep learning methods in the financial field from an empirical perspective; in reference [40] were used MLP, CNN, RNN and other models to predict stock price trends [34, 35]. Accurate rate was around 60%; in 2016, Sirignano constructed a D-dimensional spatial depth neural network model based on the probability of occurrence of real events; also some scholars used data mining methods to study stock trading trends and then analyzed their internal dynamics. Since domestic financial markets are greatly influenced by policies, domestic scholars have mainly applied deep learning to the classification algorithms of data mining [26, 41, 42]. In 2016, Zhang Chengyu used his own deep-separated neural network to study Shanghai and Shenzhen markets and reduced the forecasting error. In 2017, Zeng Zhiping studied the use of deep learning in the Shanghai and Shenzhen stock markets, increasing the accuracy rate to 90%, while Zhang Ruoxue (2018) applied machine learning to the identification of abnormal fluctuations in financial markets [43].

Neural networks also achieved some results in related research in terms of monetary policy and financial markets [44]. In 2006, Niall O'Connor and Michael G. Madden used neural networks to analyze external factors such as monetary policy and the trend of securities trading; Shin, Hyung Won, Sohn, and So Young (2007) used a combination of neural networks to study exchange rates and financial markets and proposed the exponentially weighted moving average (EWMA) model; C. Quek, M. Pasquier, N. Kumar (2008) used a recurrent neural network to construct a prediction system for options and other derivative securities [25, 45, 20]. These studies show that volatility of financial market is highly correlated with monetary policy. Zhigang Huang and Guozhong Zheng improved the BP neural network model by using the optimization algorithm, and predicted the exchange rate fluctuation trend [34, 41]. The results showed that the improved model has better applicability than EWMA model as its result is more in line with the exchange rate fluctuation characteristics [45–47]. In 2014, Nijat Mehdiyev and David Enke applied a mixed

forecasting method of data statistics and neural network to interest rate forecasting to successfully predict interest rate movements in the next three months; Study by Xiao (2017) showed that a stable monetary policy and correct interest rates enable a better function of financial markets and currency transmission mechanisms are closely related to inflation control [40, 48]. As a matter of fact, monetary policy is strongly associated with the government policy and banks. In 2017, C Borio and L Gambacorta's studies on monetary policy and bank loan recovery and J Sääski-lahti's (2018) researches on bank returns under monetary policy showed that the government's application of monetary policy can stabilize banking finance and further stabilize financial markets and then reduce systemic risks as a result [49, 50].

In general, the financial market is a noisy, nonlinear dynamic system. With the rapid development of economic and financial markets, analyzing and forecasting financial data has become a challenging task in the financial sector. However, from the current research status, there are still some shortcomings. For example, the neural network may have data over-fitting in classification, and since empirical research is often based on historical information, its future prediction and reality of finance still exist difference. The current BP-based neural network is relatively inferior in many financial risk studies, especially in the period of decline. In addition, the impact of policy on financial markets is much greater than data analysis. This paper mainly combines big data technology, applies neural network to research on the bilateral financial risk system, monetary policy expectation analysis, etc., and improves BP algorithm and then establishes WBP and TWBP model. The TWBP model improves the interim forecast through multiple cycles of the financial time effective period, and achieves it in terms of monetary policy forecasting, with a view to early warning and prevention of financial risks.

3 Bilateral financial risk system and datasets

3.1 Bilateral financial risk system

Due to the large number of financial market components, the main risk indicators affecting the financial market in the international are mainly the macro-index, currency, bond market, stock market, and other aspects, and the stock market has more and more influence on financial markets and national policies [51]. This study refers to the research results of financial institutions, financial risks and macro-control related factors, and consults the opinions of relevant experts and combines the principles of systemic, economical, and operability of evaluation indicators. The macro-index, currency, bond market, and stock market

have established corresponding first-level indicator systems, and each evaluation system has been refined. A total of 48 secondary indicators have been identified, as shown in Table 1. The detailed terminology of some detailed indicators is shown in Table 2.

In the training process of the common neural network, the weight of the general node is a computer-generated random number, and the weight value of each level node is corrected inversely according to the error between the output result and the target value. In the course of training, there are often excessive corrections or insufficient corrections. In order to avoid the imbalance of index weights and improve the training efficiency of the network, this study uses PCA to determine the relative weights of macro-index, currency, bond market, stock market and other indicators, used to intervene in the training process. The principal component analysis process is as follows:

Step (1). The data set defined as $X = (x_1, x_2, x_3, \dots, x_n)$ carries out a dimensional reduction to k .

Step (2). Use the formula $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ to calculate the average of all the samples and then normalize by $x_i = x_i - \bar{x}$

Step (3). During the preprocessing, the larger the average is, the larger the variance is. So in order to maximize the variance of the projection point, that is, the maximum eigenvalue, the eigenvector λ_u is calculated first, and then the eigenvalue \sum_u is calculated:

$$\lambda_u = \frac{1}{n} \sum_{i=1}^n ((x_i)^T u)^2 \tag{1}$$

$$\sum_u = \frac{1}{n} \sum_{i=1}^n (x_i)^T x_i \tag{2}$$

where λ_u represents feature vector, \sum_u stands for the eigenvalue u of the data set X , namely $u = (1, 2, 3, \dots, n)$, x_i signifies the i th record of the data set X , and $(x_i)^T$ embodies the matrix.

Step (4). The maximum eigenvalue is obtained. Under the condition that $u \times u^T = 1$, the maximization is $u^T \sum_u$, that is, $\sum_u - \lambda_u = 0$ is obtained. The maximized eigenvector should be the main vector corresponding to the

eigenvalue, and the maximum value of optimization is the dominant eigenvalue.

Step (5). Eigenvalues are sequenced as large to small, among which k vectors larger than others are picked as the corresponding feature vectors $u(1), u(2), u(3), \dots, u(k)$ of the output projection matrix. And such k feature vectors are used as row vectors to constitute an eigenvector matrix.

3.2 Datasets

3.2.1 Data sources

The supporting data needed for this research mainly come from financial and statistical websites and national statistical offices. In order to improve the effectiveness of basic data and reduce the complexity of data analysis, the collected data sources are from more authoritative websites and standardized databases. Therefore, the collection method is relatively simple. The main data sources and collection methods are shown in Table 3. The source is quoted from Table 3, and the source of the reference is no longer indicated. The sample data ranged from 2000 to 2018, including data on the macro-index, currency, bond market, and stock market. There are certain correlations between the collected data samples and the changes in monetary policy. The correlation is shown in Fig. 1. In the study, statistics (using SPSS) method and neural network technology (Python language) were used for training, and financial behavior was used to analyze the comparison between national monetary policy expectations and neural network models.

In Fig. 1a, chart shows the interest rate, reserves, and M0, M1, M2 trends, Fig. 1b shows the relationship between the ups and downs of Shanghai Composite Index and Shenzhen component index and the changes of interest rates and reserves, and Fig. 1c shows the relationship between the normalized gold reserve (in US dollars), net cash placement (in RMB), new deposits (in RMB), interest rates, and reserves.

Table 1 Important indicators of risk in financial markets

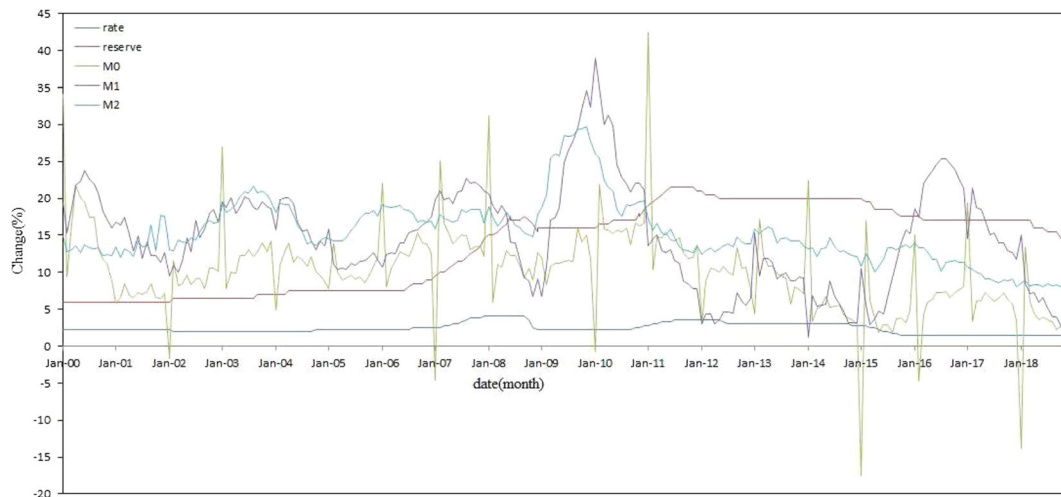
First-level indicator	Second-level indicator
Macro-index	GDP, imports and exports, balance of trade
Currency	M0 (cash in circulation), M1 (currency), M2 (currency and quasi-currency), official reserve assets, net cash placement, savings deposits, currency in circulation, exchange rate
Bond market	Net foreign assets, domestic credit, foreign debt, financial bonds
Stock market	Futures, positions, SSE, Shenzhen Stock Exchange, GEM and other indices and market capitalization

Table 2 Glossary of terms

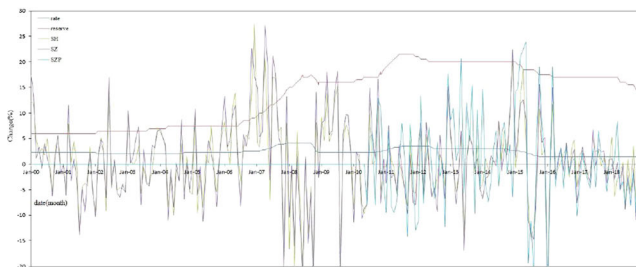
Terms	Definition	Symbols and calculations	Period	Unit
Import and export	The trade between countries, im means import, ex means export, the comprehensive value of import and export includes IM, EX and diff (the balance between import and export)	$\begin{cases} Im = \sum im \\ Ex = \sum ex \\ dif = Im - Ex \end{cases}$	Year or half a year	RMB 100 million
Balance of trade	The difference between a country's total exports and total imports In Out, indicating that income and expenditure of a country's foreign trade	$dif = In - Out$	Quarter	RMB 100 million
M0	Cash in circulation is the sum of cash in treasury of businesses rather than banks and cash that is physically used by residents. Related terms are check deposits (and bank transfer, credit card deposits) check deposits and savings deposits (including current and periodic savings deposits)	$\begin{cases} M0 = \text{cash} + \\ \text{money} \\ M1 = M0 + cd \\ M2 = M1 + sd \end{cases}$	Month	RMB 100 million
Official reserve assets	The sum of gold and foreign exchange assets held by a country to maintain balance of payments for external payments and to maintain the stability of the domestic currency exchange rate	$dif = gold + \sum fe$	MM	billions of dollars
Net cash flow	Also known as "cash withdrawal and delivery," that is, the difference between the cash debit and cash credit of the bank's external business	$delivery = cd - cc$	MM	RMB 100 million
Currency in circulation	The total amount of money that can be physically used to conduct transactions during a certain period of time, including cash, cheques, money orders, etc., that can be used as payment means	$M = \begin{cases} M1 \text{ narrow} \\ M2 \text{ general} \end{cases}$	Month	RMB 100 million
Net foreign assets	Net foreign assets consist of foreign exchange, gold deposits and the net assets of the central bank in international financial institutions	$dif = gold + \sum fe + bank$	Month	RMB 100 million
Domestic credit	The sources of credit funds and their use of credit activities in the same country	-	Month	RMB 100 million
Outstanding external debt	The total amount of outstanding external debt accumulated at a certain point in a country (or region)	$outdebt = \sum debt$	Month	RMB 100 million
Financial bond	Bonds issued by banks and other financial institutions	-	Month	RMB 100 million
Futures contract	Standardized tradable contracts with certain mass products such as cotton, soybeans, petroleum, and financial assets such as stocks and bonds	-	Day	RMB 100 million
Inventory	The market value (amount) of the currency held by the investor now accounts for the proportion of the total amount of the investment. In the futures market, open interest refers to the sum of the buy (or sell) positions before the open positions are closed	$open = \sum buy + \sum sell$	Day	
SSE	Shanghai Securities Composite Index is referred to as "Shanghai Composite Index" or "Shanghai Composite Index." Its sample stocks are all listed stocks on the Shanghai Stock Exchange, including A shares and B shares, reflecting the changes in the price of listed stocks on the Shanghai Stock Exchange. This includes turnover rate, total market value, ups and downs, etc. SZSE, GEM, etc., are similar	$\begin{aligned} \text{turnoverrate} &= \text{volum} \div \text{turnvolumm} \\ \times 100\% \text{chg} &= \text{change} \div \text{prevclose} \times 100\% \end{aligned}$	Day	

Table 3 Data sources and collection methods

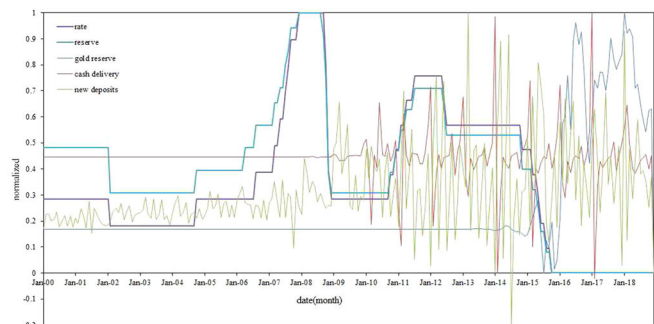
N0	Data provider	Collection method	Remarks
1	National Bureau of Statistics of China [52]	Download, web crawler	Mainly collecting Chinese macroeconomic indicators, monetary policy, etc. over the years
2	Hithink Flush Information Network Co., Ltd. (referred to as “Hithink Flush”) [53]	Interface export	Hithink Flush iFinD, part of the charge (EXCEL), easy to collect, complete information on the capital market
3	CSMAR [54]	Interface Download	Has been organized into normative big data database in financial filed



(a) Interest rate, reserves, and M0, M1, M2 trends



(b) Interest rate, reserves, and stock index trends



(c) Interest rate, reserves, and gold reserve, net cash placement, new deposits trends

Fig. 1 Examples of financial data to be predicted

3.2.2 Data preprocessing

The data collected come from different data sources, and therefore there is no uniform standard for data integrity and data format. In order to make the collected data more stable and effective in the training of the neural network, we need to further extract and process the collected data. The processing time period is in units of months. The processing steps are as follows:

(1) Missing value processing

First filter null, invalid, zero, and likely invalid values and analyze the cause of missing values and then replace the

missing values with the most likely values. This data set mainly adopts (Likewise Deletion), (Mean Imputation), (Regression Imputation), etc., and detecting data missing one by one in time order. Use o , t to represent the original data set, in the processed data set, o_i represents the i th data in the total amount n of the original data set. When the value of o_i is “-” or “NULL,” it is considered that there is a missing value, When the value of o_i is 0 and the original data set of o does not have a value of 0, that is, the zero value collected is unreasonable.

Likewise Deletion: $t = o - \sum o'$, refers to in exchange for the integrity of the data set at the expense of the original

data. Many scholars try not to adopt this method when researching. For this dataset, since the GEM is not listed during the period from 2000 to 2010, it is in a blank trading range. Therefore, this part of the data cannot use other filling methods. Code to delete the SQL is *delete from [bank2nonnull] where [opening price] is null*, and this code can be run in the data set compiled in this study by sql server 2012.

Mean Imputation: $t = o_{1i} + \sum_{x=i}^{x=j} \bar{o}_{ij} + o_{jk} + \dots + o_{mn}$, $\bar{o} = \frac{\sum o_i}{n}$ represents mean. This study extends the mean to two forms: the first one $\bar{o}_{ij} = \frac{\sum_{x=i}^{x=j} o_x}{j-i+1}$, which indicates the imputation of the mean between the *i*th and *j*th. The second form combines time series, increments or decrements the data difference between the current period and the previous period, expressed as $\bar{o}_{ij} = o_i + \frac{o_i - o_{i-1}}{j-i+1}$, which is especially suitable for quarterly statistics when it needs to be split into monthly data, such as GDP and foreign debt.

Regression Imputation: $t = o_p + o_q$, o_q represents the missing sets, the rest data is expressed as o_p . This study uses the most typical linear regression fill, the predicted value of the linear regression model as the estimated value of the missing value, the *k*th data missing value is expressed as $o_{pk} = a_0 + \sum_{x=i}^{x=j} a_i o_{ik} + \varepsilon_k$. When the regression relationship between the variables is significant, the method will become more realistic, such as the imputation of foreign net assets, bonds, and domestic credit in September and November 2001.

(2) Data normalization

The statistics of financial data is mainly time-based. First, the time line should be standardized. For the convenience of analysis, the processing time period is in units of months, while the stock market-related data are based on working days. At the end of the month, the data are calculated by the interval, the monthly turnover rate, the price increase, the market value, etc.; the units between the data are inconsistent, resulting in excessive differences between the data, which is not conducive to analysis; as a result, the normalization of the ranks is required.

Data self-normalization (normalization of column): $Col_i = Col_i / Col_{max}$. That is, the current value is divided by the maximum value.

The normalization of the data rows is consistent with the normalization processing method of the column. The processing is to make the difference between the data of the receiving nodes of the neural network smaller, and the normalization processing of the rows is also the same. $Row_i = Row_i / Row_{max}$.

(3) Continuous data consolidation

First, sort data in chronological order. For the policy release time and stock market time which may be out of

sync, such as the impact of the workday, the deviation affects the correction code as follows:

```
update [reserves] set [stock market time]
=DATEADD (day, Deviation number of days,
CTIME)
where DATEADD (day,1, CTIME) in
(select [Time] from [bank2nonnull]) and [stock market
time] is null
```

Then you have to make the data in time. For example, if the data rate between the interest rate and the stock market is connected, the date of the month is as follows:

```
select ([The current deposit changes]+[The current
loan changes])/2,t.* from three t left outer join [rate] r
on datediff (month, t.[Time], r. stock market time)
= 0
update [reserves] set [stock market time] = b.[stock
market time]
from [reservesold] b
where [reserves].[CTIME] = b.[CTIME]
```

3.2.3 Training case design

In order to reduce the error in the backpropagation, the output node value (monetary policy) is optimized, that is, the financial domain expert adjusts the label on the output node to adapt to different neural network training. The adjustment case and adjustment variant are as follows: $rate_i$, res_i indicate the interest rate, the value of the *i*th period of the reserve, and $rate_i - rate_{i-1}$ indicates the change value of the interest rate.

(1) Case A

The interest rate and the change in reserves are used to measure the intensity of policy control. 0 means no adjustment. The larger the number is, the stronger the adjustment will be. The adjustment of interest rate is higher than the reserve, and output nodes are labeled as [0–9]. The maximum value is represented by max. The minimum value is represented by min, [] indicates rounding, and the label output of case A is:

$$L_a = \begin{cases} 9, & (rate_i - rate_{i-1} \neq 0 \text{ and } res_i - res_{i-1} \neq 0) \\ 4 + \left[\frac{(rate_i - rate_{i-1}) \times 4}{\max(rate_i - rate_{i-1}) - \min(rate_i - rate_{i-1})} \right], & (rate_i - rate_{i-1} \neq 0) \\ \left[\frac{(res_i - res_{i-1}) \times 4}{\max(res_i - res_{i-1}) - \min(res_i - res_{i-1})} \right], & (res_i - res_{i-1} \neq 0) \\ 0, & (rate_i - rate_{i-1} = 0 \text{ and } res_i - res_{i-1} = 0) \end{cases} \tag{3}$$

where $rate_i - rate_{i-1} \neq 0$ represents the interest rate change of the i th time point and last time ($i - 1$) (The impact of the interest rate change on the financial market is greater than the reserve modification.) In like manner, the formula $res_i - res_{i-1} \neq 0$ represents the reserve change of the i th time point and last time ($i - 1$). When the results of such two formulas are not zero, it means the interest rate and the reserve have changed and the financial market indicators of the i th time point have undergone great changes. On the contrary, when the result is zero, it means the currency policy has stayed the same.

This approach is recognized by a majority of financial experts. The different levels of policy are believed more relevant to financial risks.

(2) Case B

Output nodes are labelled as 0, 1, 2, 3, that is, interest rate and reserve change as 3, interest rate change as 2, reserve change as 1, no change as 0. When computer experts need to improve case A and the data set is not big enough, case A training is relatively poor, and the changes are in Eq. (4):

$$L_b = \begin{cases} 3, & (rate_i - rate_{i-1} \neq 0 \text{ and } res_i - res_{i-1} \neq 0) \\ 2, & (rate_i - rate_{i-1} \neq 0) \\ 1, & (res_i - res_{i-1} \neq 0) \\ 0, & (rate_i - rate_{i-1} = 0 \text{ and } res_i - res_{i-1} = 0) \end{cases} \tag{4}$$

where $rate_i - rate_{i-1} \neq 0$ and $res_i - res_{i-1} \neq 0$ are the same as the ones in Eq. (3). When their results are not zero, $L_b(i)$ is labeled as 3. Conversely, $L_b(i)$ is labeled as 0. It also shows that the impact of the interest rate change on the financial market is greater than the reserve modification.

(3) Case C

Output nodes are labelled as 0, 1, that is, interest rate or reserve change as 1, no change as 0. Some financial experts believe that this adjustment is too simple to distinguish the effect of monetary policy, calculated in Eq. (5):

$$L_c = \begin{cases} 1, & (rate_i - rate_{i-1} \neq 0 \text{ or } res_i - res_{i-1} \neq 0) \\ 0, & (rate_i - rate_{i-1} = 0 \text{ and } res_i - res_{i-1} = 0) \end{cases} \tag{5}$$

where $rate_i - rate_{i-1} \neq 0$ and $res_i - res_{i-1} \neq 0$ are the same as the ones in Eq. (3). On the contrary, when the result is zero, it means the currency policy has stayed the same.

(4) Case D

It is mainly based on raw data such as interest rates and reserves and has two branches based on data processing.

(4.1) Case D1

Combine the current value of the interest rate and the current value of the reserve, that is, adding original value:

$$L_{d1} = rate_i + res_i \tag{6}$$

where $rate_i$ and res_i represent the currency policy and reserve of the i th time point, respectively.

(4.2) Case D2

Current value of interest rate and reserve as output node:

$$L_{d2-rate} = rate_i, L_{d2-res} = res_i \tag{7}$$

where $rate_i$ and res_i represent the currency policy and reserve of the i th time point, respectively.

(5) Case E

Financial experts believe that the adjustment of monetary policy is more conducive to judging the direction of the financial market. Therefore, the value of interest rate and reserve changes on the label is used as the output node, and 0 means no change, as shown in Eq. (8):

$$L_e = \begin{cases} rate_i - rate_{i-1}, & (rate_i - rate_{i-1} \neq 0) \\ res_i - res_{i-1}, & (res_i - res_{i-1} \neq 0) \\ 0, & (rate_i - rate_{i-1} = 0 \text{ and } res_i - res_{i-1} = 0) \end{cases} \tag{8}$$

where $rate_i - rate_{i-1}$ and $res_i - res_{i-1}$ represent the change value of interest rate and reserve between the i th and ($i - 1$)th time point(last time). If the result is zero, it means the currency policy has not changed.

4 Neural network model

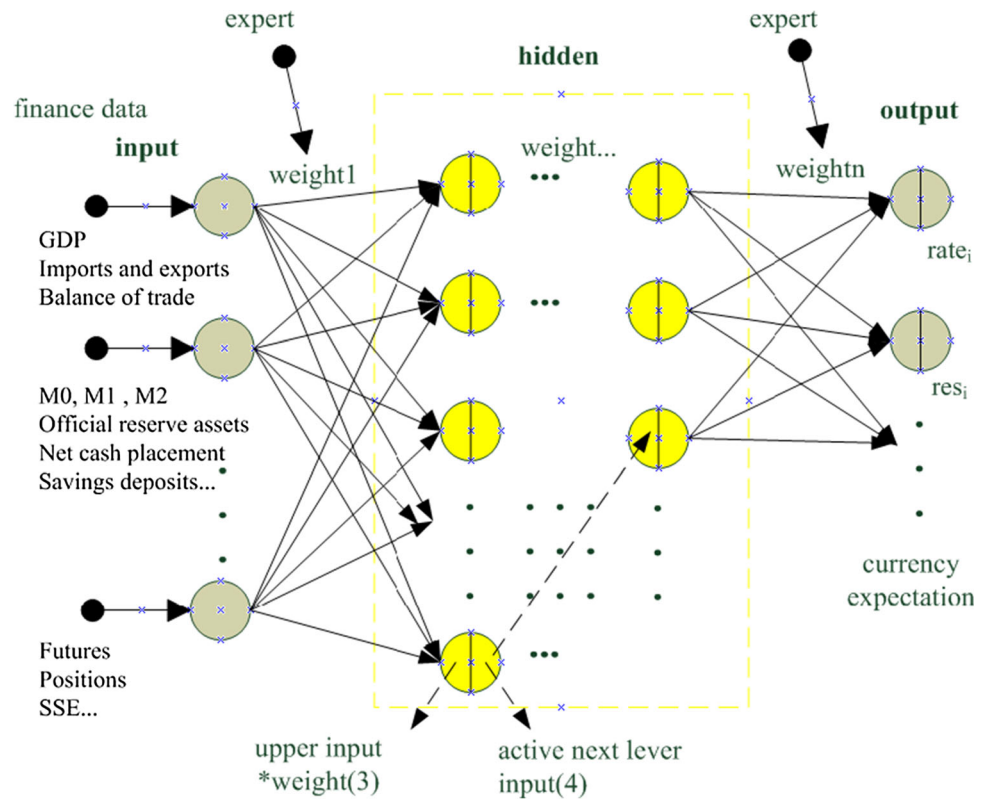
4.1 BP model

According to the data set collected by the bilateral financial risk system, with the month data as input, the input nodes transmit to the hidden layer with random weight and then hides the layer through the middle of the neural network, and the number of layers and the number of nodes are uncertain (all equal to or greater than 1). Verify that the training output is consistent with the currency expectations. Inconsistent errors are inversely corrected and continuously trained to obtain better results [29, 20]. The original transmission process of BP neural network is shown in Fig. 2. The neural network model runs as follows:

Step (1). Read the financial data set and get the input matrix I

Step (2). Establish weight matrix W_{ij} , where i is the number of current layer nodes. The first time is the number of initial input nodes, j is the number of nodes in the next layer, and the number of nodes in the last layer is the number of nodes in the output layer.

Fig. 2 Neural network model. From the hidden layer to the output layer, each node is divided into two parts. The first half is the cloud computing result of the input and weight of the previous layer, and the second half is the input of the next layer by the activation function



$$X = W \cdot I \tag{9}$$

where W represents a weight matrix and I stands for the input nodes.

Step (3). Combine the data and apply the activation function to the next layer. X is the combined data. It also needs to smooth and activate X . The activation form is diversified, and its effect is different. The commonly used activation functions are shown in Table 4.

$$O = \text{activefunction}(X) \tag{10}$$

where x is the result of input nodes times weight, which is activated and used as the input for the next layer.

The active function of the first and last layers in this paper applies sigmoid, and the hidden layer applies ReLU active function.

Step (4). At the hidden layer, repeat Step 3 until it outputs to the last layer, the output layer.

Step (5). Calculate error, compare output layer with target value

$$e_{\text{output}} = \text{target} - O_{\text{output}} \tag{11}$$

where target stands for target node and O stands for the output result for the last layer.

Step (6). Backpropagation error. Assigning the respective errors based on input weights is the most common

practice in current neural networks. In the case of back-propagation, the input weight matrix W_{ij} needs to be transposed W^T , and repeat the propagation error until the first layer of the network.

$$e_{n-1} = e_n \cdot W^T \tag{12}$$

where n represents nodes in the n th layer (the input layer is the first layer), e_n represents deviation, and $(W_{n-1})^T$ stands for weight matrix transposing.

Step (7). Weight correction. The weight is corrected by the error. The learning rate a of the neural network is introduced, that is, the decreased gradient.

$$W_{ij} = W_{ij}(\text{old}) - a \cdot \frac{\partial E}{\partial W_{ij}} \tag{13}$$

where i represents the n th layer (the input layer is the first one), j signifies the j th node in the i th layer, W_{ij} and $W_{ij}(\text{old})$ stand for modified weight value and weight value before correction for the j th node in the i th layer, a represents learning rate, $\partial E, \partial W$ are the differentiation of deviation and weight in the i th layer.

Step (8). Repeat step 2 to 7 until cyclic generation designed ends. The steps of the BP initialization are shown in the algorithm.

Algorithm 1 BP initialization

Inputs: *input_nodes*: Number of Financial data;
hidden_nodes: Default hidden nodes;
level: Number of hidden layers;
output_nodes: Number of Monetary Policy;
learning_rate: Learning rate.

Outputs: *self*: The class after initialization.

- 1: self.inodes, onodes, level \leftarrow *input_nodes*, *output_nodes*, *level*;
- 2: self.nodes \leftarrow [0 for i in range (level+2)], initialize input, output, hidden nodes matrix;
- 3: self.weights \leftarrow [(0 \times 2) for i in range (self.level+1)], initialize weight matrix;
- 4: for i in range (level): Set hidden layer's nodes and weights
- 5: Assignment of each layer nodes, self.nodes[i+1] \leftarrow *hidden_nodes*;
- 6: Fitting weights by normal distribution,
- 7: self.weights[i] $\leftarrow \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(self.weights[i]-\mu)^2}{2\sigma^2})$,
- 8: weights matrix scale \leftarrow (self.nodes[i+1],self.nodes[i]);
- 9: if int(self.nodes[i+1]) < =0: self.nodes[i+1] \leftarrow *hidden_nodes*;
- 10: self.nodes[0] \leftarrow *input_nodes*;
- 11: self.nodes[self.level+1] \leftarrow *output_nodes*;
- 12: self.weights[self.level] $\leftarrow \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(self.weights[self.level]-\mu)^2}{2\sigma^2})$,
- 13: last layer's weights matrix scale \leftarrow (self.nodes[self.level+1], self.nodes[self.level]);
- 14: self.lr \leftarrow *learning_rate*;
- 15: return self.

In algorithm 1, the structure of neural network is initialized with the number of input nodes, hidden nodes and output nodes, respectively. Three for loop statements are used to define the nodes and weights of each layer, and the random initial weights are assigned.

4.2 WBP model

In the research of the financial field, many studies have shown that the correlation between the original data has obvious strong correlation. The detailed indicators in the bilateral financial risk system constructed in this paper have a relatively direct and close relationship with monetary policy, but their importance is slightly different. Experts' opinions vary. Use the PCA method to obtain the initial weight of each indicator $W_{out} = weight^T$, for weight

correction intervention. The WBP model is established as shown in Fig. 3 based on the weighting of the input layer to the hidden layer and the hidden layer to the output layer. For the uncertain factors such as the number of layers inside the hidden layer and the number of nodes, we still process them by BP model. The adjustment of different cases and the setting of weights have certain influences in the training process, and they have certain adaptability in different periods. The external weight is introduced in Eq. (9) and is changed to $X = W_{out} \cdot I$ at the time of initialization, and in the above in Eq. (13) W_{ij} needs to re-interpolate externally, that is, $W = W_{out} \cdot W_{ij}$. The weight, error, and the output between the input layer and the hidden layer are represented by W_1 , e_1 , and O_1 . The weight of the first layer is improved in Eq. (14):

Table 4 Characteristics the used activation functions [55, 56]

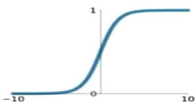
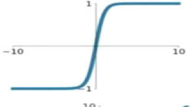
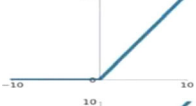

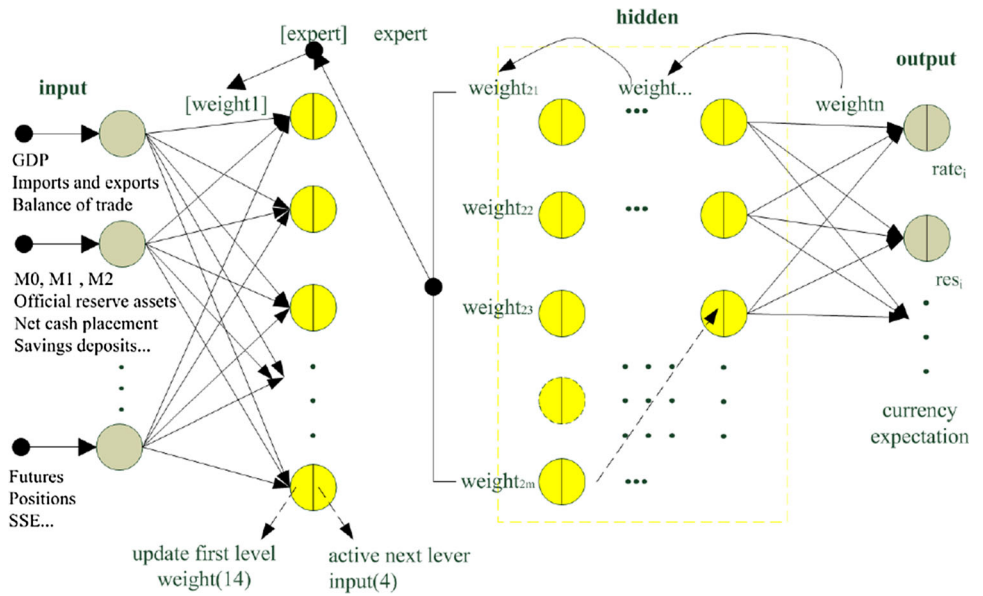
Active function	Formula	Activation form	Description
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$		Map data to (0, 1), less used for the hidden layer
Tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$		Map data to (-1, 1)
ReLU	$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x < 0 \end{cases}$		Also known as a modified linear unit. Higher efficiency, but when the input is negative, the gradient disappears
Leaky ReLU	$f(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{if } x < 0 \end{cases}$		The gradient disappears when correcting the problem that input is negative

Fig. 3 The proposed WBP model. Expertise refers to expert weights, external weights are added on the basis of BP, the first layer of input and feedback are affected by external weights, and other layers are still corrected by the computer



$$W_1 = a \times \left(\frac{e_1 \times O_1}{1 - O_1} \times I \right)^T \tag{14}$$

where $W_1, e_1,$ and O_1 represent the weight, deviation, and the output between input layer and hidden layer. $\hat{\delta}$ stands for gradient, and I represents input nodes.

cannot directly operate with W_{ij} , so the correction weight intervention is during the backpropagation to the input nodes. The initialization is reinitialized in the forward calculation, sub-algorithms such as 10.1 to 10.4. The detail steps of the training process of the proposed WBP model are shown in the algorithm.

Algorithm 2 The training process of the WBP model

```

Inputs: input_list: Financial data, input list;
          target_list: Monetary policy, output list;
          level: Number of hidden layer;
          startweight: External weights.
Outputs: weights: Modified weight set.
1: Transpose input and target matrix, inputs, targets  $\leftarrow$  input_listT, target_listT;
2: Initialize matrix, ins, outs, errors  $\leftarrow$  [(0 × 2) for i in range (level+1)];
3: for i in range (1,level+1):
4:   ins[i]  $\leftarrow$  weights[i] × outs[i - 1];
5:   Applying active function
6:   if i=1 or i=level:
7:     outs[i]  $\leftarrow$   $\frac{1}{1 + e^{-ins[i]}}$ ;
8:   else
9:     outs[i]  $\leftarrow$   $\begin{cases} ins[i], & \text{if } ins[i] > 0; \\ 0, & \text{if } ins[i] < 0; \end{cases}$ 
10: errors[level]  $\leftarrow$  targets - outs[level];
11: for i in range (self.level,0,-1):
12:   Error back propagation, errors[i-1]  $\leftarrow$  self.weights[i]T × errors[i];
13:   Correction weight, weights[i]  $\leftarrow$  self.lr × ( $\frac{errors[i] \times outs[i]}{1.0 - outs[i]}$ ) × outs[i - 1]T;
14: External weight intervention, inputs  $\leftarrow$  inputs × self.startweight;
15: weights[0]  $\leftarrow$  self.lr × ( $\frac{errors[0] \times outs[0]}{1.0 - outs[0]}$ ) × inputsT;
16: return weights.
    
```

The WBP model pays more attention to the external intervention process of weight correction, while W_{out} relies on the calculation of experts and PCA on the one hand and the weight of each indicator on the other hand. Since the number of hidden layer nodes of the BP model is inconsistent with the number of input nodes, the hidden layer

In algorithm 2, use financial data as input list and monetary policy as output list. The output list is also called target node value. The data combined with weights assigned by financial experts are trained in neural network. The first for loop statement mainly realizes the forward operation of neural network, and the second for loop statement mainly

realizes the backpropagation of error correction and modifies the weight of each layer node. According to the training needs, we can also carry out a layer of peripheral circular statements to achieve the epochs of training (general training of six epochs).

4.3 TWBP model

Most of the collection of financial data is in time order. Many scholars have studied the combination of time series and neural networks in the financial field, such as neural network time series prediction models for financial data. Monetary policy is more affected by the latest economic environment, and the farther the previous period is, the smaller the impact is. Therefore, the above neural network model is further changed according to the characteristics of financial data and based on the model of time effective period. The selected feature indicators were postponed from January 2000 to December 2009 or later. Firstly, train the WBP model neural network to learn the relationship between monetary policy and risk system indicators, and then apply a certain time cycle to neural network. Repeat the training. Iterate over subsequent time periods, allowing

the financial data closer to the expected period to be trained. We propose a time effective period model based on WBP, denoted as TWBP. The specific TWBP model change structure is shown in Fig. 4.

The TWBP model needs to segment and train the input on the basis of Eq. (9) $X = W_{out} \cdot I$; the base represents the WBP basic training, the offsets represent the offset training step, and the loop represents the cycle.

$$X_i = \begin{cases} W_{out} \cdot I_i, & (i \leq base) \\ W_{out} \cdot I_{base+offset+j}, & (i \geq base \text{ and } j = loop) \end{cases} \quad (15)$$

The model within the scope of basic training base is the same as the method in Eq. (9). But when i is beyond base, designated step size offset and neural network cycle loop(0, 1, 2, . . . , k) are used as the new input. Basic training sample length subtracted from total sample length is neural network cycle loop(0, 1, 2, . . . , k). Subsequent deviation training will be repeated continuously with the passage of time to ensure the latest time data have more opportunities to be trained. The algorithm of cyclic neural network based on time period offset is expressed as shown in the algorithm.

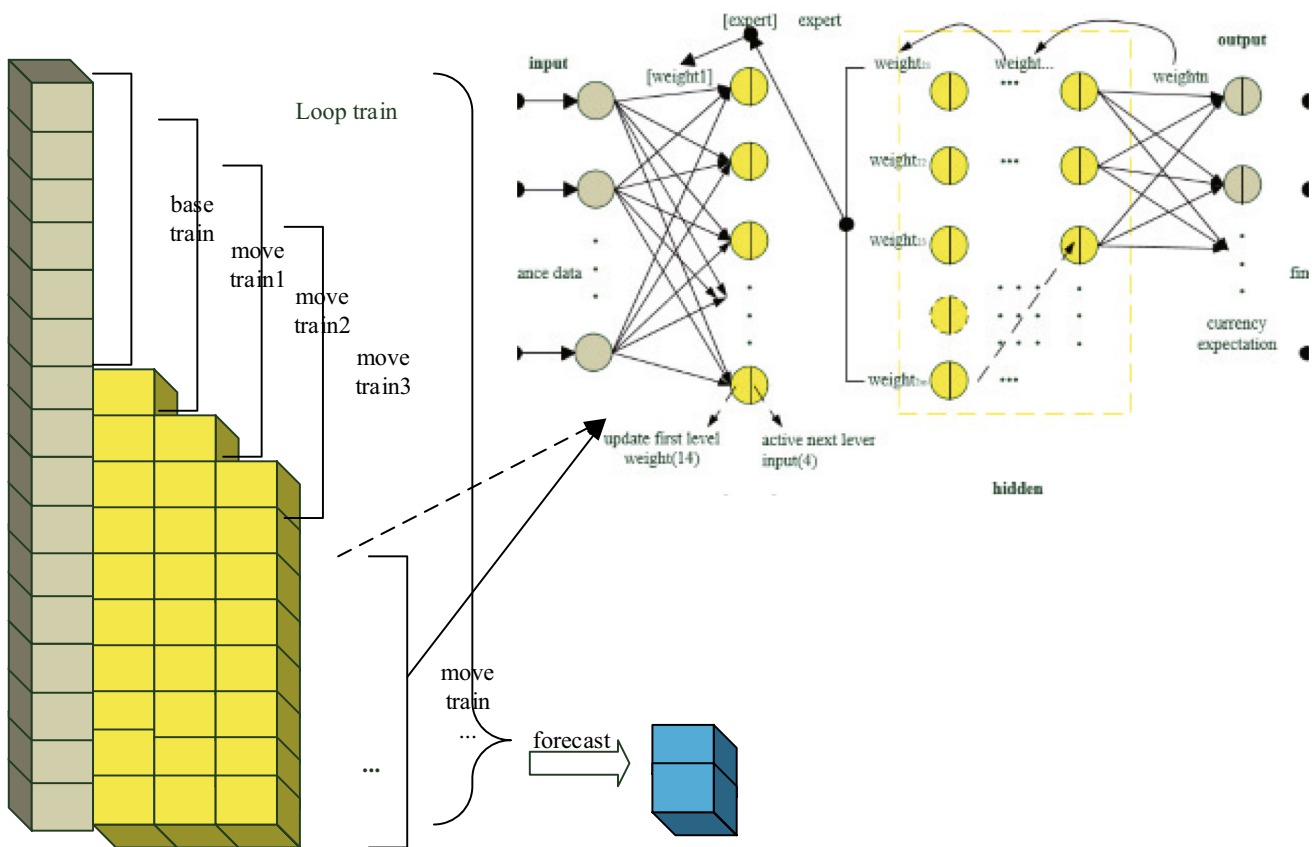


Fig. 4 Cyclic neural network based on time period offset. First, complete a relatively large cycle training and develop a certain basic cycle time. Each cycle will perform WBP training for the specified number of iterations, then move down a minimum time period and

predict it, and then complete the basic learning. Compare the results generated by network that completed basic learning with next prediction result, so that the network can learn further until it moves down to the bottom, and finally completes the prediction function

Algorithm 3 TWBP cycling training

Inputs: *data_list*: Training data;
nmove: Initial training cycle;
loop: The cycle of the circulating neural network;
epochs: Number of iterations;
min: Minimum time period;
startweight: External weights.

Outputs: *r_array*: Training results comparison list.

```

1: r_array ← [(0) × 4] for p in range (loop);
2: for i in range (loop):
3:   Downward training, train_list ← data_list[i: i+nmove-1];
4:   Minimum test time period, test_list ← data_list[i+nmove: i+nmove+min];
5:   Multiple cyclic training network
6:   for e in range (epochs):
7:     for record in train_list:
8:       Data format and initialize, target ← record.split(',').[0];
9:       inputs ← record.split(',').[1:];
10:      External weight intervention
11:       $inputs \leftarrow \begin{cases} startweight \cdot inputs[i], & \text{if } (i \leq nmove) \\ startweight \cdot inputs[i + e + record], & \text{if } (i \geq nmove \text{ and } record = loop) \end{cases}$ ;
12:      Use WBP train, WBP(inputs, target);
13:      Compare and record test results
14:      for record in test_list:
15:        target ← record.split(',').[0];
16:        inputs ← record.split(',').[1:];
17:        inputs ← inputs × startweight;
18:        Used to compare training results, output ← self.query (inputs);
19:        Set two-dimensional array, r_array[i] ← nmove+i+1, output, all_values[0];
20: return r_array.
```

In algorithm 3, using five for loop statements to fulfill TWBP training, the former half is the training loop (train WBP) and the latter half is the test loop (query). By cyclic operation step by step, the training data and test data of the neural network are set, and the data are divided into the input and output lists before entering the neural network. At the same time, the input node is initialized by expert weight. *r_array* mainly realizes the storage of test data and actual data, which facilitates subsequent data output and analysis.

5 Experiment analysis

5.1 Experimental setting

The software used in this study uses the MyEclipse 2017 CI development tool, and the development language is python 3.7. The data set processing storage form includes SQL SERVER 2012 database and csv text. Because the data set is not very large and the neural network does not

need too much in the training effective hidden layer test, the requirements can be met within 10 layers, so the hardware requirements are relatively low. The experiment and recommendations are configured as shown in Table 5:

5.2 Case

The neural network will be affected by the learning rate, the number of hidden layers, the hidden node distribution and the number of output nodes during the training process. Some settings such as excessive hidden layers will invalidate the experimental results. According to the data set, the six cases of Table 4 are combined with the BP, WBP and TWBP models. Based on the test of stable learning rate, hidden layer number and hidden node distribution, 18 kinds of experiments can be obtained. As a result, the comparative analysis is as follows.

Table 5 Description of the hardware experiment environment

Hardware	Experimental configuration	Related influence
CPU	i5 7500 3.4G Quad core four thread	CPU has a significant impact on TWBP operations, while memory and SSD have essentially no effect
RAM	DDR4 2400 16G	
SSD	Intel 256G	

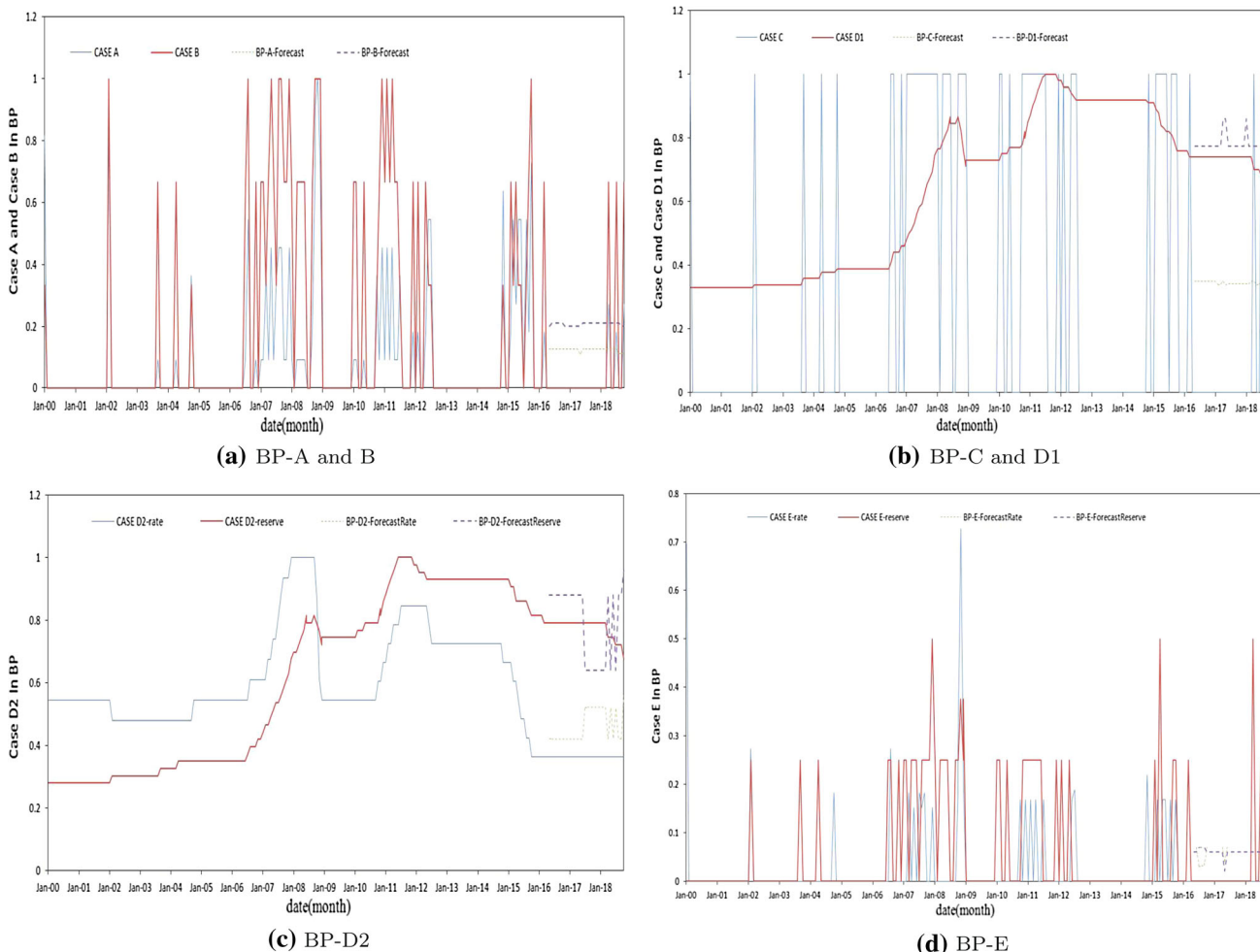


Fig. 5 Results of different cases in BP model

5.2.1 Performance evaluation of BP model

The experimental results of the BP model on different cases are shown in Fig. 5. The results of the model training are linear and parallel. Different cases are different because the BP neural network output nodes are different, and the pros and cons of the case cannot be judged directly by predicting data. Judging the accuracy of the results by the cumulative error size is suitable for most prediction methods. Assuming n-stage prediction, the actual value and predicted value of the *i*th period are represented by T_i, P_i , and the prediction accuracy Exact is expressed as:

$$Exact = \frac{\sum_{i=1}^n |T_i - P_i|}{n} \times 100\% \tag{16}$$

Absolute value of actual value T_i subtracted from P_i is accumulated and then the average of the accumulation is taken as the overall accuracy.

Through the error analysis, the error from cases A, B, C is getting larger and larger, that is, the accuracy is lower and lower. Therefore, the less the output of the label in the

setting process, the more unfavorable the correction weight of the model training combination. Case C is the least desirable solution, and case D1 has the least error in training results. This also applies to the WBP and TWBP models, so when the raw data is simply processed or not processed, the neural network is more likely to find its relevance. The manual participation in the modified label seems to have some rationality, but it has already destroyed the hidden association between the data. Case D2 and E both separated the interest rate and reserve, so the case D2 data training effect of raw value (not the original data of non-financial data, but the normalized values) is better than the case E change value, which is consistent with the hidden relationship between the original data. However, we also noticed that case E training is more stable, the interest rate and reserve error are very close, and the interest rate forecast in case D2 is obviously not as good as the reserve forecast. It can also be seen that the interest rate change occurs less frequently, and the interest rate is lower. It has a considerable impact on the adjustment of financial markets.

Therefore, the monetary policy of all countries will be very cautious in adjusting interest rates. For example, the Fed’s previous interest rate hike cycle has a significant impact on the global financial market.

In Fig. 5a, the prediction error range of cases A and B neural network is between 0.1 and 0.17; between 0.21 and 0.46, and case C is between 0.34 and 0.66. More than 50% of the error has indicated that the case C is not suitable for neural network training. In Fig. 5b, case D1 shows an improvement, as shown in the figure, the prediction coincides with the actual value, and the error is within 0.06. In Fig. 5c, the reserve prediction accuracy rate in case D2 is higher than the interest rate, while in Fig. 5d, the reserve prediction reserve ratio in case E is lower than the interest rate.

5.2.2 Performance evaluation of WBP model

The experimental results of the WBP model on different cases are shown in Fig. 6. The results of the training of the model are also linear and parallel, and the error

performance is different. The interest rates for cases C and D2 are the least desirable. However, from the overall prediction effect, the error of the WBP model is reduced by 50% compared with the BP model; in particular, the cases A and E have more than 80% improvement. Except for case D2, the interest rate prediction error adjusted by this case is larger. We found that the training results of cases A and D1 have the smallest error, and the error can be controlled within 10%. The initial weight comes from the financial experts’ prediction of the indicators. Monetary policy is closely related to the financial market, and the adjustment of weight is more inclined to the macro-control results produced by policy changes.

In Fig. 6a, WBP model has a great consistency in terms of the case error compared with the BP model, but the training results of WBP model are more accurate (Case A prediction error range is between 0.03 and 0.12) and the error interval is narrower (Case B error is stabilized between 0.31 and 0.36); in Fig. 6b, although case C is improved in WBP model training, of which its maximum error drops to 0.57, the inaccuracy rate is still too high. In

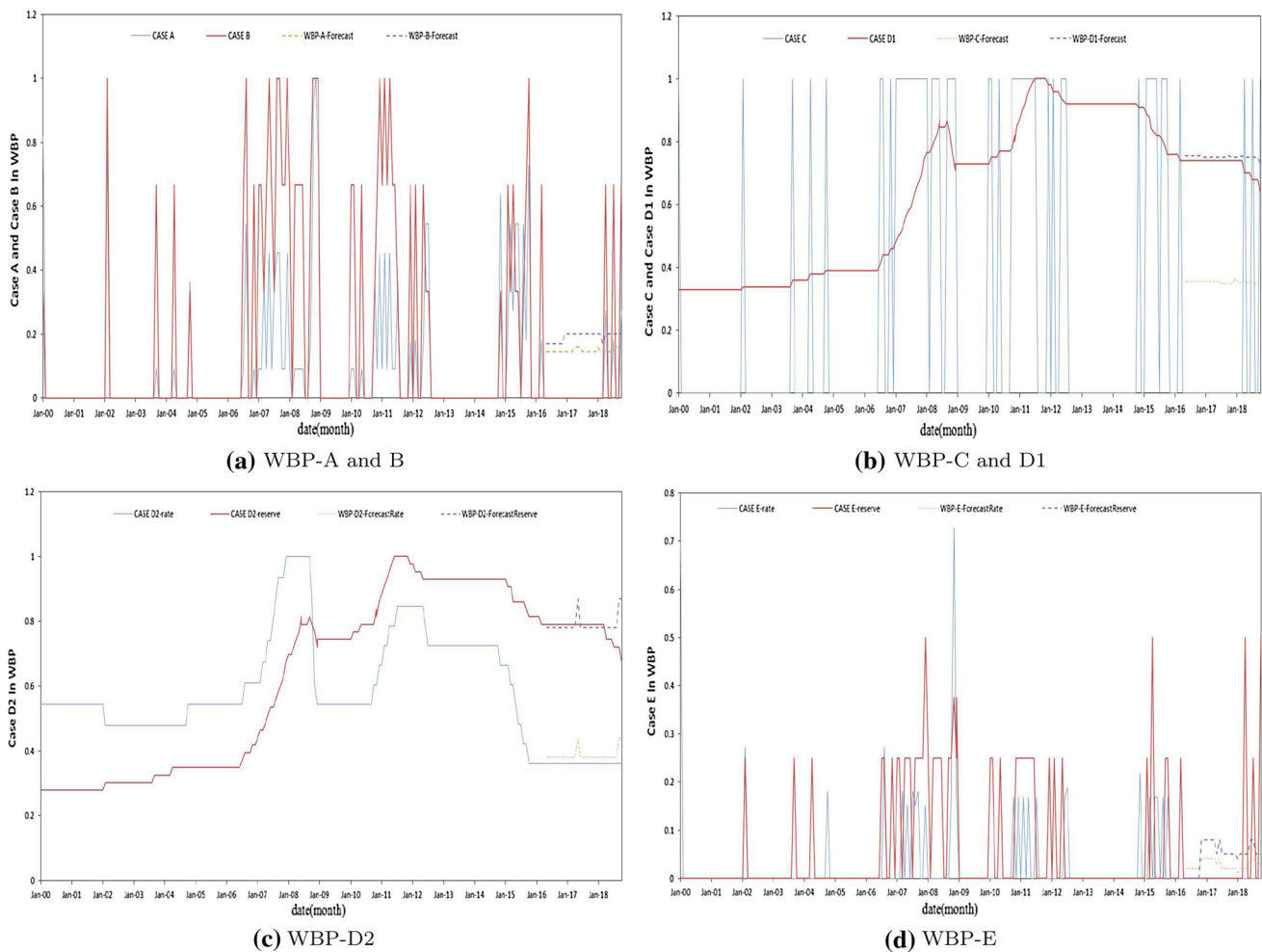


Fig. 6 Results of different cases in WBP model

Fig. 6c, the prediction accuracy of reserve in case D2 is still higher than that of interest rate, and the maximum error is 0.11 and 0.28, respectively; In Fig. 6d, the result is the opposite, the reserve prediction reserve ratio in case E is lower than that of interest rate, and compared with the BP model, the maximum error increased by 0.07 and 0.18, respectively, and the minimum error of the reserve was greatly improved, dropping to 0.01.

5.2.3 Performance evaluation of TWBP model

The experimental results of the TWBP model in different scenarios are shown in Fig. 7. The results of the training of the model are no longer linear and parallel, but the index values corresponding to the cycle are close to change. Solutions with an error of less than 5% include cases A, D1, D2, and E. Case C is still the least ideal solution, so that we can conclude that when the output result label is too single, the hidden association of the data is not easy to learn during training.

As shown in Fig. 7, the TWBP model has a great improvement in terms of minimum error. Except case C in Fig. 7b, the minimum error of other cases is less than 0.03, and the error shows a particularly significant decrease with time. The training results of the TWBP model are also more stable, the error gradually decreases with time, and the learning effect of the neural network is quite obvious. In Fig. 7b, case D1 and in Fig. 7c, case D2 have three precise predictions that are equal to the actual values. Among them, the minimum error of prediction of interest rate in case E in Fig. 7d is 0.03, and the minimum error occurrence frequency accounts for 32.7% of the overall prediction (107 predictions, 35 times minimum errors, and the rest 24 errors are within 0.03).

5.3 Experimental results of different models

The results of different models are compared, as shown in Fig. 8. The upper part of the figure is the training result of each model, and the lower part is the normalized data value

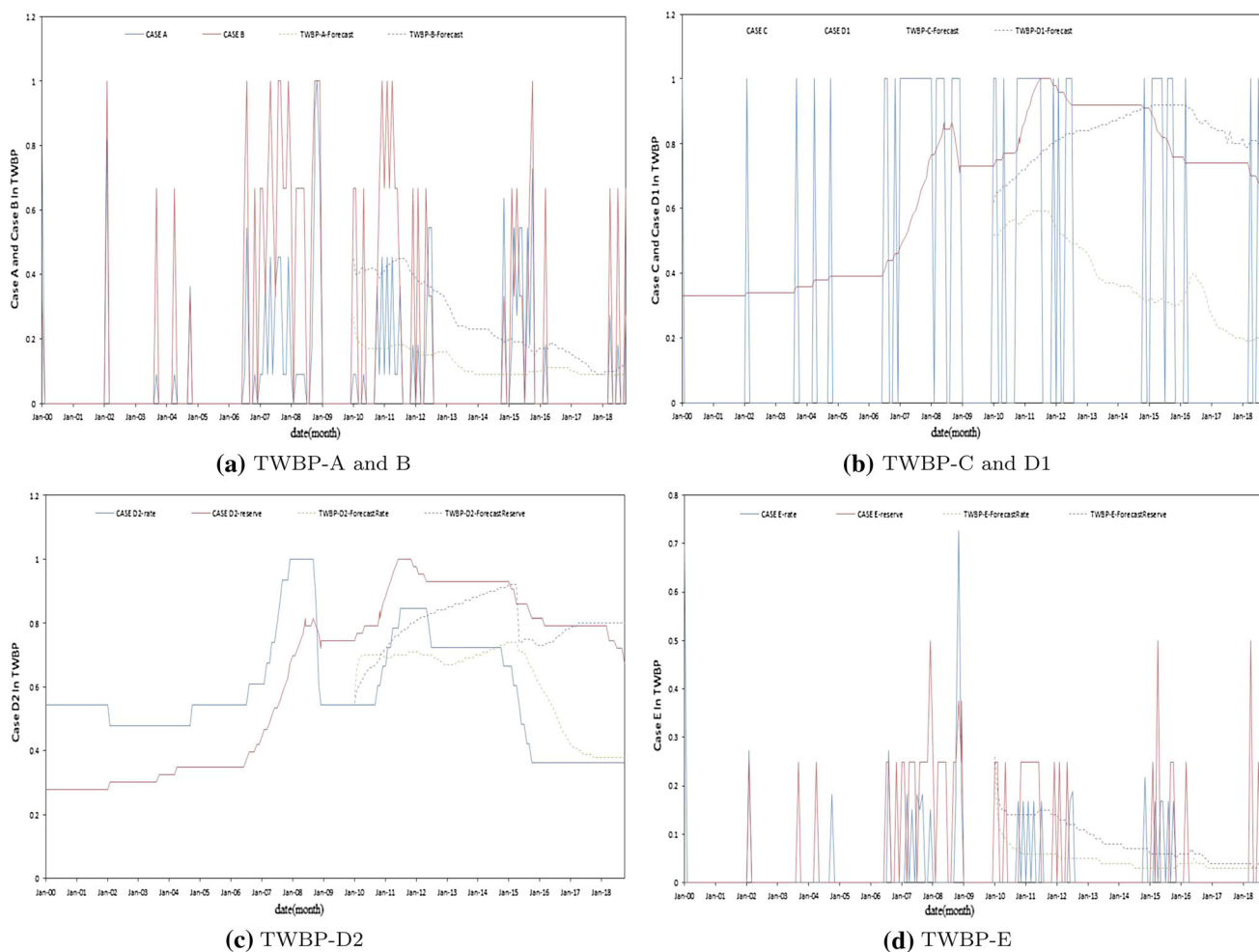


Fig. 7 Results of different cases in TWBP model

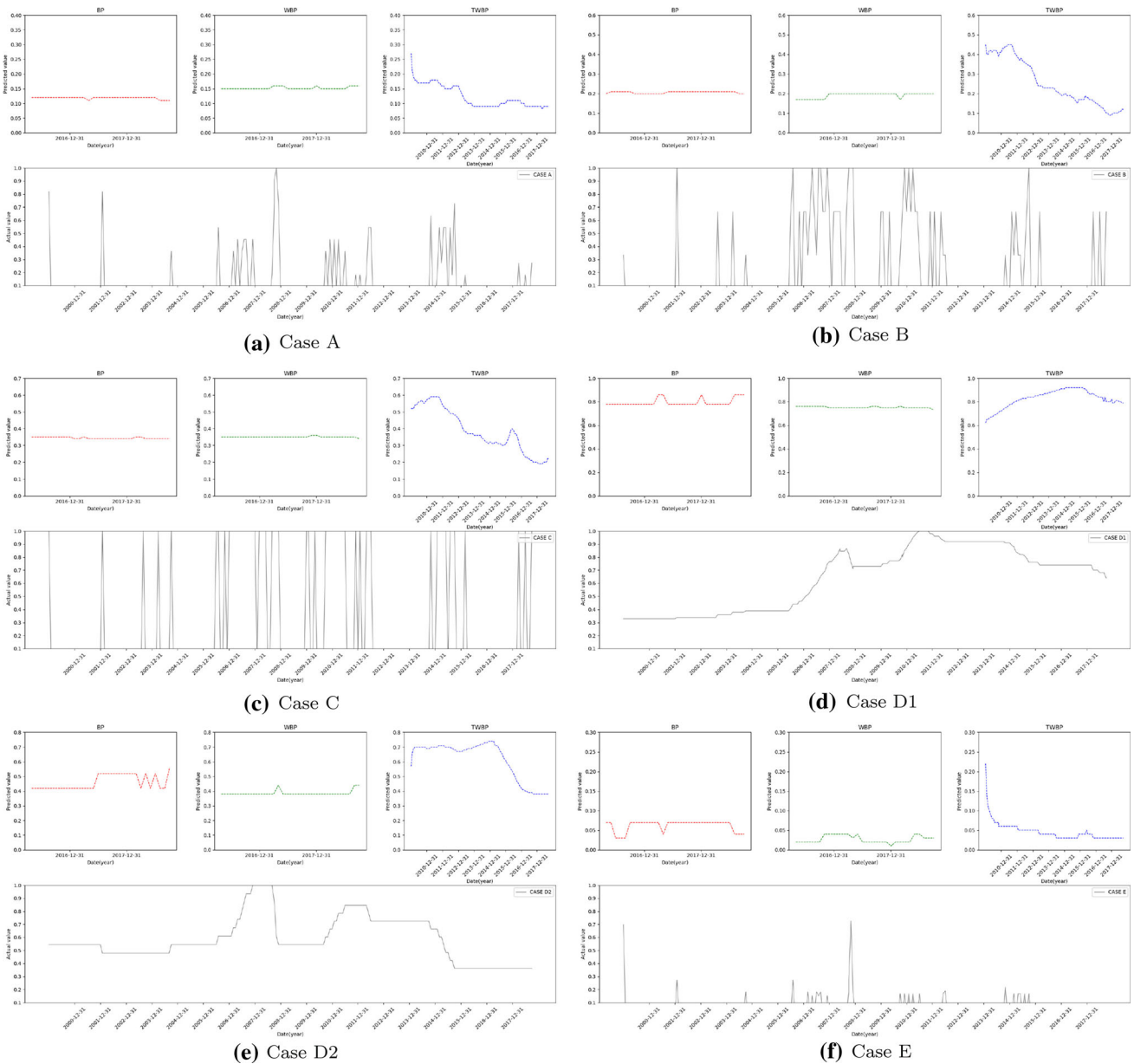


Fig. 8 Experimental results of different models

of cases. By comparison, BP and WBP models have a short predictive period and are not as accurate as the TWBP model. Therefore, for the weighted composite neural network aiming for monetary policy, TWBP model is more effective. In terms of error, TWBP is better than WBP, and WBP is better than BP. However, due to the long prediction period, TWBP has a large rate of prediction error in the early stage, and even deviates. For example, cases D1 and E are obviously deviated. Since the training period of BP and WBP models (15 years) is more than half of that of TWBP model (10 years), the prediction of TWBP model is not as good as that of BP and WBP models. Therefore, the training data set of neural network is more effective when it

is larger. As the prediction error is continuously corrected, the TWBP model significantly improves the accuracy of the prediction, while the BP and WBP models do not incorporate the periodic exercise of the financial time series. The prediction cannot track the error and continuously improve, and its accuracy is only more effective in the early stage of prediction.

As shown in Fig. 8, the TWBP model has a long prediction period, and the initial prediction is not as good as the BP and WBP models, but with the passage of time, the error is getting smaller and smaller, that is, the accuracy is getting higher and higher, for example, in subgraph (d), the accuracy of BP and WBP is 89%, while the accuracy of

TWBP model is only 81% at the beginning, but it can reach 95% afterward. Other subgraphs also show higher prediction rates of TWBP than that of BP and WBP models. Taking the TWBP model as an example, we compare the errors between different cases. Case C has the largest error mutation, while cases D1, D2, and E are relatively stable.

In Fig. 9, in general, cases A, B, and C are not as stable as cases D1, D2, and E. In the early, middle, and late stages, cases D2 and E are more and more stable.

5.4 Error and performance analysis

From the experimental results of different models, it can be seen that the TWBP prediction effect is significantly improved. According to the training error of each

scheme shown in Fig. 10, the training efficiency of the TWBP model is relatively low. By setting the number of hidden layers 3, 30, 300 to compare, we found that as the hidden layer increases, the time required for TWBP training is also increasing. The average calculation is based on the training time (s), and the performance comparison is shown in Fig. 10.

In Fig. 10, neural network comparison under the setting of 5 layers of hidden layer, learning rate of 0.2 in Fig. 10a: taking the error average as a comparative analysis, the case D1 scheme is better than expected to be stable at around 0.03. The TWBP model has the best effect. The average cases A, D1 and E errors are less than 0.05, and the WBP model is the second. The D2 training result rate is opposite to the case E prediction; in Fig. 10b, the hidden layer

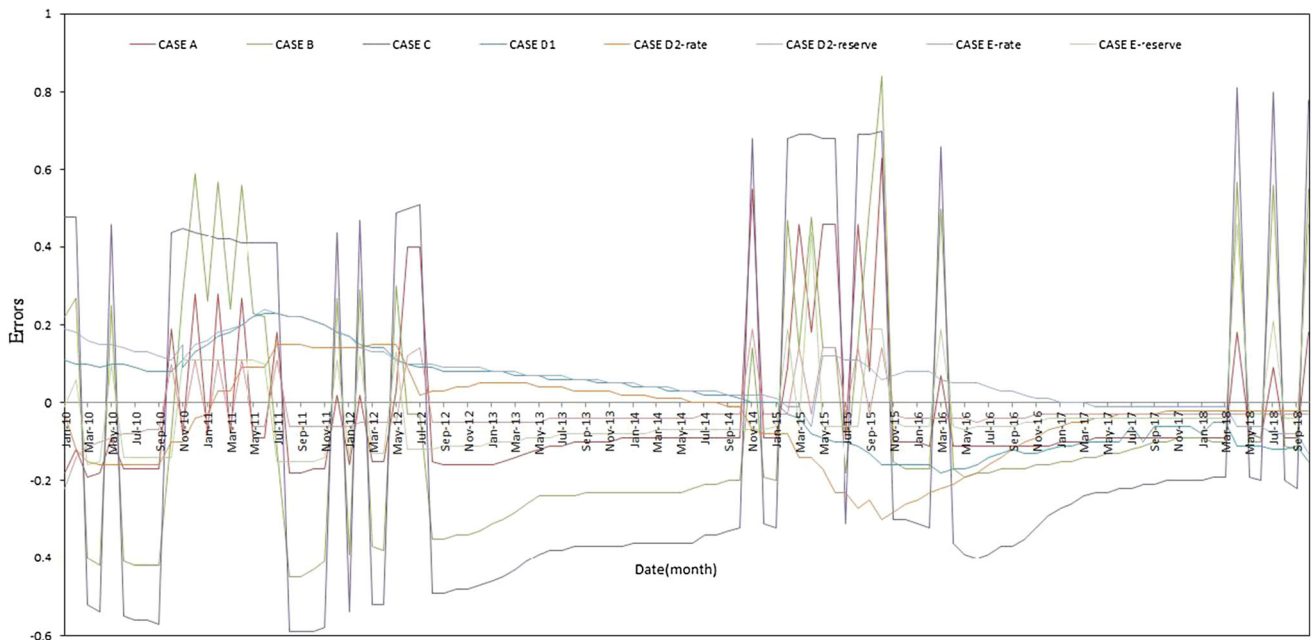


Fig. 9 Error comparison and test of different cases of TWBP model training

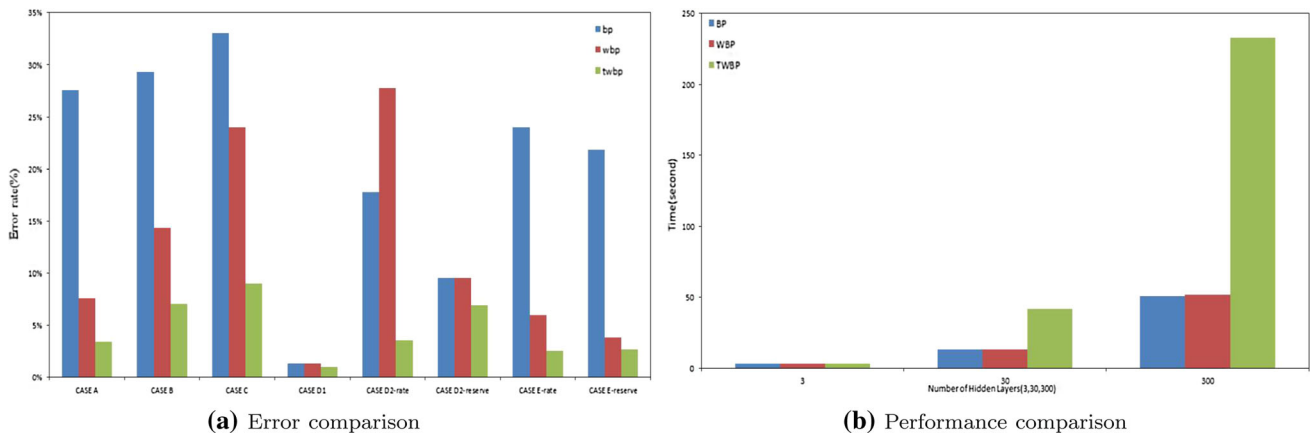


Fig. 10 Error and related performance comparison

number is used to test 3, 30, and 300, respectively. The hidden layer number of setting 3 is basically about 3 seconds to complete the training, while in the hidden layer of 300, the TWBP model is used. The time consumed is more than five times that of the other two models. The training efficiency is sacrificed when the TWBP model improves the training effect. When the data set is larger, the model still needs further performance improvement.

6 Conclusion

This paper established the indicator dataset through PCA and carried out related preprocessing such as data missing value, data normalization, data continuous merging, and then designs six training schemes. The algorithm proposed WBP and TWBP models and experimented with different training programs. From the perspective of neural network learning, the characteristics of the data feature set are favorable for time series prediction, which can better fit the actual value, while the artificial intervention of the data feature will reduce the prediction effect. Experiments show that the important indicators of financial market risk extracted by this study are in line with training expectations and can be related to the relationship between monetary policy and financial market risk. Combined with the current state of operation of China's financial market, this paper proposes some recommendations. When the data of important indicators of financial market risk suddenly change, especially the indicators with high weight of experts, the state needs to timely adopt macro-control monetary policy, and the important indicators of financial market risk after macro-control can be effectively controlled. To be worth raising, the interest rate adjustment of monetary policy is greater influence than the reserve adjustment. Therefore, the interest rate adjustment that stimulates the economy is conducive to the stability of the short-term financial market, especially in the securities market. The effectiveness of the correlation mechanism between it and interest rates indicates the need to improve the policy system of interest rate regulation to improve the securities market environment. Financial market performance economic indicators have largely reflected in the securities market. Therefore, the government should strengthen supervision over the securities market and formulate a reasonable monetary policy. The expected monetary policy can promote the development of the economy and prevent the occurrence of financial risks.

Acknowledgements This work was supported social science research base of finance and accounting research center of Fujian province, in part by asset evaluation research project of Fujian social science planning and research base under Grant JXY201801-08 and JXY201801-05, in part by the "Internet+" virtual simulation

experimental teaching platform for the key project of the young and middle-aged teachers in Fujian province under Grant JZ180190, in part by the active project reporting mode under the background of cloud network in the major project of Fujian province social science base in 2018 Research under Grant FJ2018JDZ014, in part by the project of Fujian institute of technology under Grant GB-RJ-17-58.

Compliance with ethical standards

Conflict of interest The author Minrong Lu is from Fujian Jiangxia University, and he declares that he has no conflict of interest.

References

- Black SE, Devereux PJ, Lundborg P (2017) On the origins of risk-taking in financial markets. *J Financ* 72(5):100–105
- Riccioni J, Cerqueti R (2018) Regular paths in financial markets: investigating the Benford's law. *Chaos Solitons Fractals* 107:186–194
- Lin EMH, Sun EW, Yu MT (2018) Systemic risk, financial markets, and performance of financial institutions. *Ann Oper Res* 262(2):579–603
- Lengnick M, Wohltmann HW (2016) Optimal monetary policy in a new Keynesian model with animal spirits and financial markets. *J Econ Dyn Control* 64:148–165
- Dafermos Y, Nikolaidi M, Galanis G (2018) Climate change, financial stability and monetary policy. *Ecol Econ* 152:219–234
- Platen E, Rebolledo R (1996) Principles for modelling financial markets. *J Appl Probab* 33(3):601–613
- Li Kenli, Tang Xiaoyong, Li Keqin (2014) Energy-efficient stochastic task scheduling on heterogeneous computing systems. *IEEE Trans Parallel Distrib Syst* 25(11):2867–2876
- Antipov A, Meade N (2002) Forecasting call frequency at a financial services call centre. *J Oper Res Soc* 53(9):953–960
- Li C, Chiang TW (2013) Complex neurofuzzy ARIMA forecasting—a new approach using complex fuzzy sets. *IEEE Trans Fuzzy Syst* 21(3):567–584
- Kayalar DE, Küçüközmen CC, Selcuk-Kestel AS (2017) The impact of crude oil prices on financial market indicators: copula approach. *Energy Econ* 61:162–173
- Yuming Xu, Li Kenli, Jingtong Hu, Li Keqin (2014) A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. *Inf Sci* 270:255–287
- Ganda F (2019) The environmental impacts of financial development in OECD countries: a panel GMM approach. *Environ Sci Pollut Res* 5:11–15
- Jian ZH, Zhu BS, Shuang LI (2012) Dynamic inflation target, money supply mechanism and China's economic fluctuation: A DSGE-based analysis. *Chin J Manag Sci*
- Yuming Xu, Li Kenli, He Ligang, Zhang Longxin, Li Keqin (2015) A hybrid chemical reaction optimization scheme for task scheduling on heterogeneous computing systems. *IEEE Trans Parallel Distrib Syst* 26(12):3208–3222
- Chen J, Li K, Bilal K, Metwally AA, Li K, Yu PS (2018) Parallel protein community detection in large-scale ppi networks based on multi-source learning. *IEEE/ACM Trans Comput Biol Bioinf*. <https://doi.org/10.1109/TCBB.2018.2868088>
- Chen C, Tiao GC (1990) Random level-shift time series models, arima approximations, and level-shift detection. *J Bus Econ Stat* 8(1):83–97
- Einav L, Levin J (2014) Economics in the age of big data. *Science* 346(6210):124–130

18. Zhang XPS, Wang F (2017) Signal processing for finance, economics, and marketing: concepts, framework, and big data applications. *IEEE Signal Process Mag* 34(3):14–35
19. Araújo RDA (2012) A morphological perceptron with gradient-based learning for Brazilian stock market forecasting. *Neural Netw* 28:61–81
20. Chung H, Shin KS (2010) Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability* 10:3765
21. Li Kenli, Yang Wangdong, Li Keqin (2015) Performance analysis and optimization for SPMV on GPU using probabilistic modeling. *IEEE Trans Parallel Distrib Syst* 26(1):196–205
22. Chen Jianguo, Li Kenli, Tang Zhuo, Yu Shui, Li Keqin (2017) A parallel random forest algorithm for big data in spark cloud computing environment. *IEEE Trans Parallel Distrib Syst* 28(4):919–933
23. Lazer D, Kennedy R, King G (2014) Big data. the parable of Google Flu: traps in big data analysis. *Science* 343(6176):1203
24. Chiu MC, Pun CS, Wong HY (2017) Big data challenges of high-dimensional continuous-time mean-variance portfolio selection and a remedy. *Risk Analysis* 37(8):1532
25. Chen Jianguo, Li Kenli, Deng Qingying, Li Keqin, Yu Philip S (2019) Distributed deep learning model for intelligent video surveillance systems with edge computing. *IEEE Trans Ind Inf*. <https://doi.org/10.1109/TII.2019.2909473>
26. Yu L, Chen H, Wang S (2009) Evolving least squares support vector machines for stock market trend mining. *IEEE Trans Evol Comput* 13(1):87–102
27. Ebrahimpour R, Nikoo H, Masoudnia S (2011) Mixture of MLP-experts for trend forecasting of time series: a case study of the tehran stock exchange. *Int J Forecast* 27(3):804–816
28. Chen MY, Chen BT (2015) A hybrid fuzzy time series model based on granular computing for stock price forecasting. *Inf Sci* 294(2):227–241
29. Chen Jianguo, Li Kenli, Bilal Kashif, Zhou Xu, Li Keqin, Yu Philip S (2018) A bi-layered parallel training architecture for large-scale convolutional neural networks. *IEEE Trans Parallel Distrib Syst* 30(5):965–976. <https://doi.org/10.1109/TPDS.2018.2877359>
30. Chen J, Li K, Huigui R, Bilal K, Li K, Yu PS (2018) A periodicity-based parallel time series prediction algorithm in cloud computing environments. *Inf Sci*. <https://doi.org/10.1016/j.ins.2018.06.045>
31. Bengio Y, Delalleau O (2009) Justifying and generalizing contrastive divergence. *Neural Comput* 21(6):1601
32. Cao Z, Long C, Chao Z (2015) Spiking neural network-based target tracking control for autonomous mobile robots. *Neural Comput Appl* 26(8):1839–1847
33. Dahl GE, Sainath TN, Hinton GE (2013) Improving deep neural networks for LVCSR using rectified linear units and dropout. In: *IEEE international conference on acoustics*
34. Qian B, Rasheed K (2007) Stock market prediction with multiple classifiers. *Appl Intell* 26(1):25–33
35. Fratianni M, Marchionne F (2013) The fading stock market response to announcements of bank bailouts. *J Financ Stabil* 9(1):69–89
36. Patwary EU, Lee JY, Nobi A (2017) Changes of hierarchical network in local and world stock market. *J Kor Phys Soc* 71(7):444–451
37. Wu X, Lu H (2010) Exponential synchronization of weighted general delay coupled and non-delay coupled dynamical networks. *Comput Math Appl* 60(8):2476–2487
38. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436
39. Song Y, Lee JW, Lee J (2018) A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction. *Appl Intell* 49:897–911
40. Pang X, Zhou Y, Pan W (2018) An innovative neural network approach for stock market prediction. *J Supercomput* 1:1–21
41. Asadi S, Hadavandi E, Mehmanpazir F (2012) Hybridization of evolutionary Levenberg-Marquardt neural networks and data pre-processing for stock market prediction. *Knowledge-Based Syst* 35(15):245–258
42. Yu-Hui T, Wei K, Shou-Ning Q (2010) Research on stock quotation based on improved rough lattice model. *IEEE* 3:442–445
43. Yoo PD, Kim MH, Jan T (2005) Financial forecasting: advanced machine learning techniques in stock market analysis. *IEEE*, 1–7
44. O'Connor N, Madden MG (2006) A neural network approach to predicting stock exchange movements using external factors. *Knowledge-Based Syst* 19(5):371–378
45. Chen Yuedan, Li Kenli, Yang Wangdong, Xiao Guoqing, Xie Xianghui, Li Tao (2019) Performance-aware model for sparse matrix-matrix multiplication on the sunway TaihuLight supercomputer. *IEEE Trans Parallel Distrib Syst* 30(4):923–938
46. Maknickiené N, Maknickas A (2013) Financial market prediction system with Evolino neural network and Delphi method. *Physiologia Plantarum* 14(2):403–413
47. Jie W, Wang J (2017) Forecasting stochastic neural network based on financial empirical mode decomposition. *Neural Netw* 90:8–20
48. Xiao Guoqing, Li Kenli, Li Keqin (2017) Reporting 1 most influential objects in uncertain databases based on probabilistic reverse top-k queries. *Inf Sci* 405:207–226
49. Desai VS, Bharati R (2010) The efficacy of neural networks in predicting returns on stock and bond indices. *Decis Sci* 29(2):405–423
50. Xiao G, Li K (2019) CASpMV: a customized and accelerative SPMV framework for the sunway TaihuLight. *IEEE Transactions on Parallel and Distributed Systems* pp. 1–1. <https://doi.org/10.1109/TPDS.2019.2907537>
51. Lin A, Shang P, Zhou H (2014) Cross-correlations and structures of stock markets based on multiscale MF-DXA and PCA. *Nonlinear Dyn* 78(1):485–494
52. Statistical data: National bureau of statistics of china (nbsc). URL <http://www.stats.gov.cn/>
53. iFind: Hithink flush information network co., ltd. (ifind). URL <http://www.51ifind.com/>
54. CSMAR: Tai'an information technology co., ltd. (csmar). URL <http://www.gtarsc.com/>
55. Leshno M, Lin VY, Pinkus A (1991) Original contribution: multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Comput* 6(6):861–867
56. Mandic DP, Chambers JA (1999) Relating the slope of the activation function and the learning rate within a recurrent neural network. *Neural Comput* 11(5):1069–1077

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.