A study of bank failure risk based on neural network model

Mengyi Kang[1,*], Kang Shi[2], Menghan Ma[3]

[1]School of Mathematics and Statistics, Anhui Normal University, Wuhu 241003, Anhui, China
[2]School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang 050043, Hebei, China
[3]School of Architecture and Construction, Shijiazhuang Tiedao University, Shijiazhuang 050043, Hebei, China
*Corresponding author

Abstract: This paper focuses on the analysis of bank failure causes i.e. prediction. Based on the neural network model, we use principal component analysis and fuzzy C-mean cluster analysis algorithm to achieve the evaluation of bank efficiency, the mining of bank insolvency causes, and the prediction of bank failure risk. By considering the literature search and production method together, we classify the 64 indicators of the indicator set into input indicator layer and output indicator layer. In the process of data analysis, we found that some of the higher weights do not objectively reflect the causes of bank failures. In order to avoid simply selecting 5 indicators from the 64 indicators, we used principal component analysis to reduce the dimensionality, set the number of extracted factors to 5, and consider the bank failure factors comprehensively through the 5 selected principal components, and calculate the percentage weights of the eigenvalues calculated for each principal component each year to complete the calculation of each bank's score. Whether a bank fails or not has a strong nonlinear relationship with 64 indicators, and considering that some of the data present exponential relationship, logarithmic relationship, less pure linear relationship between the data, and the indicator dimension is too large. Therefore, we decided to build a model based on BP neural network model, and continuously reduce the error by forward, backward transmission.

Keywords: Entropy weight method, principal component analysis, BP neural network, fuzzy C-value clustering analysis

1. Introduction

Banks are the core of the modern economy and the foundation of social credit. Since its birth, it has taken on the social responsibility of making social capital flow and played an important role in promoting social progress. In order to deeply analyze and predict the causes of bank failures, the following analysis is given for the causes of bankruptcy and prediction of bankruptcy trends of insolvent banks, using the data on the operation of existing or failed banks in Polish banks for five years as an example[1].

First, the data are divided into two types of data: input and output; can we continue to analyze the pre-processed data and calculate the efficiency of each bank; based on the calculated efficiency of each bank per year, we determine the bankruptcy decomposition line for each bank per year, and then determine the bankruptcy decomposition line per year and the total bankruptcy decomposition line for five years.

Secondly, we analyze the changes in the data of 64 indicators of failed banks, find the common points of the changes in indicators, and analyze the causes of bankruptcy; whether we can find the five most critical factors that lead to bankruptcy. In order to comprehensively and objectively consider the causes of bank failures, we use principal component analysis to comprehensively consider the influencing factors of the 64 indicators and reduce the dimensionality, determine the weight relationship of each principal component in each year based on the eigenvalues calculated for each principal component in each year, and then calculate the score of each bank.

Finally, we set up a BP neural network model, using the 64 indicators of the original data as the training set and the scores calculated in Problem 2 as the output set, and obtained the prediction results of the neural network model by iterating the neurons in the hidden layer, and improving the fit of the

model by changing the parameters to improve the accuracy of the model prediction.

2. Model Methodology and Construction

2.1 Analyzing the Bank Failure Problem Model

In the principal component analysis model, the principal component y has:

$$
\begin{cases}
y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p = a_1^T x \\
y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2p}x_p = a_2^T x \\
\ldots\ldots\ldots \\
y_p = a_{p1}x_2 + a_{p2}x_2 + \ldots + a_{pp}x_p = a_p^T x
\end{cases}
\tag{1}
$$

Abbreviated as:

$$
y = Ax, A = (a_1, a_2, \ldots a_p)^T
\tag{2}
$$

Following the requirement that the covariance matrix of y should be a diagonal matrix, there are:

$$
Cov(y) = Cov(Ax) = ACov(x)A^T = A\sum A^T = \Lambda = diag(\lambda_1, \lambda_2, \ldots \lambda_p)
\tag{3}
$$

Since the covariance matrix $\sum$ of x is usually a symmetric positive definite matrix, according to the fundamental theorem of linear algebra, there must exist a set of unit orthogonal vectors $a_1, a_2, \ldots ap$ constituting an orthogonal matrix A such that (3) holds, and if the characteristic roots of $\sum$ are also ordered by size $\lambda_1 \geqq \lambda_2 \geqq . \geqq \lambda_p \geqq 0$, then $a_1, a_2, \ldots a_p$ are their corresponding unit orthogonal eigenvectors[2].

The linear combination of the ith principal component is obtained by filtering the eigenvalue factors and calculating the degree of their contribution to the original variables for the selected principal components. The formula for calculating the sample-related linear matrix of the X matrix is as follows:

$$
R = \frac{\sum\limits_{k=1}^{n} \left(x_{ki} - \overline{x_i}\right)\left(x_{kj} - \overline{x_j}\right)}{\sqrt{\sum\limits_{k=1}^{n} \left(x_{ki} - \overline{x_i}\right)^2 \sum\limits_{k=1}^{n} \left(x_{kj} - \overline{x_j}\right)^2}}
\tag{4}
$$

Calculate the eigenvalues of R:

$$
tr(R) = \sum_{k=1}^{p} \lambda_k = p
\tag{5}
$$

Model building is performed on this basis.

(1) By analyzing the data, 64 indicators were identified (x1, x2, ..., x64)

(2) Denote $\lambda_i = \left(\overline{\lambda_{i1}}, \overline{\lambda_{i2}}, \ldots, \overline{\lambda_{ip}}\right)^T (i = 1, 2, \ldots, 64)$ as an observation of a 64-dimensional random variable.

The mean vector $\overline{\lambda} = \frac{1}{n}\sum\limits_{i=1}^{n} x_i = \left(\overline{\lambda_1}, \overline{\lambda_2}, \ldots \overline{\lambda_p}\right)^T$ as the estimate of $E(\lambda) = \mu$.

The covariance matrix of X $S = \frac{1}{n-1}\sum\limits_{i=1}^{n}(\lambda_i - \overline{\lambda})(\lambda_i - \lambda)^T$ is used as the estimate of $Cov(x) = \Sigma$.

(3) Compute the eigenvectors of S

The eigenroots of the covariance matrix S are calculated by MATLAB programming, which in turn yields the variance contribution of each principal component in order to determine the number of principal components.

S can be considered as the covariance matrix $S = \dfrac{1}{n-1}\sum\limits_{i=1}^{n}\left(\lambda_i - \bar{\lambda}\right)\left(\lambda_i - \bar{\lambda}\right)^T$ as the estimate of

$Cov(x) = \sum$.

The principal component coefficient a_1, a_2,...,a_m are calculated from $ACov(x)A^T = \Lambda$.

(4) Calculate the cumulative contribution of the eigenvalues λ_i:

$$b_j = \frac{\lambda_j}{\sum\limits_{k=1}^{m}\lambda_k}\left(j = 1,2,..m\right) \tag{6}$$

(5) Based on the output of the principal component coefficient matrix, the first principal component can be obtained as:

$$y_i = a_1\lambda_1^* + a_2\lambda_2^* + ...a_{64}\lambda_{64}^* \tag{7}$$

2.2 Bank Failure Risk Prediction Model

The efficiency of bank's failure risk is related to 64 indicators. In order to comprehensively consider the influence of each indicator among the bank failures, the data of 64 full indicators selected in this paper are used to predict the failure risk of each bank; the comprehensive index score of each bank is an important factor to measure whether a bank is in the risk of failure, so the risk index of each bank is used as the output layer and the solution model is established based on the BP neural network prediction model[3].

In the BP neural network prediction model, when a set of designs is provided to the network, it is mainly composed of input forward propagation and error backward propagation, and the weights are constantly modified by the network error, and the weight matrix is modified according to the actual output and the expected output to reduce the structural error of the neural network. The structure of the model is shown in Figure 1 .
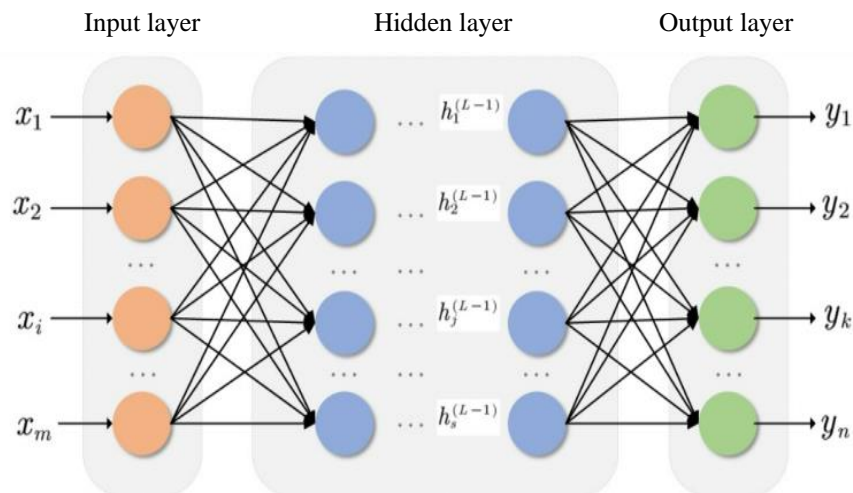

Figure 1: Multi-layer neural network structure diagram

Usually a multilayer neural network consists of three layers of neurons, the first layer is called the input layer, the last layer is called the output layer, and the middle layer is the hidden layer.

The basic components of a multilayer neural network are neurons, and the model of a single neuron is as Figure 2:
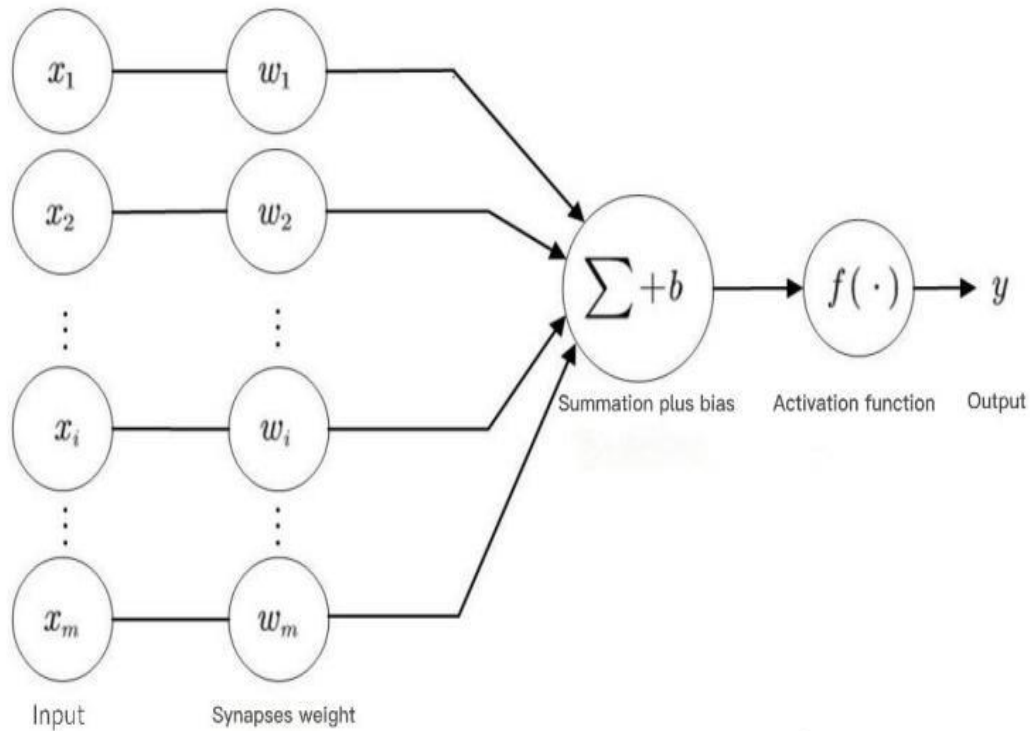
Figure 2: Models of single neurons

(1) Create input and output layer vectors

Input layer vector X=(xi1,xi2,... ,xi64)

Output layer vector Y=(y1,y2,... ,y2)

(2) Forward transmission phase

Take a sample Pi, Qi from the sample and input Pi into the neural network;

Calculate the error measure Ei and the actual output $O_i = F_l\left(...\left(F_2\left(F_1\left(P_iW^{(2)}\right)W^{(2)}\right)...\right)W^{(L)}\right)$.

The weight values W$^{(1)}$, W$^{(2)}$,... ...W$^{(L)}$ are each adjusted once, and this cycle is repeated until $\sum$i<ε.

(3) Backward propagation stage - error propagation stage

Calculating the difference between the actual output O$_p$ and the ideal output Q.

Adjusting the output layer weight matrix with the errors in the output layer.

$$E_i = \frac{1}{2}\sum_{j=1}^{m}\left(Q_{ij} - O_{ij}\right)^2 \tag{8}$$

(4) Formation of a layer-by-layer transmission process

The error measure of the network about the whole sample set: the error Ei is used to estimate the error of the direct leading layer of the output layer, and then the error of the leading layer of the output layer is used to estimate the error of the further layer, and so the error estimates of the other layers are obtained.

And these estimates are used to implement modifications to the weight matrix, forming the process of transferring the errors exhibited at the output to the output in a step-by-step manner along the direction opposite to the output signal.

$$E = \sum_i E_i \tag{9}$$

Modeling on this basis.

(1) Selecting the activation function

To analyze the data of this problem, in order to introduce the nonlinear factor, so that the model can better approximate the nonlinear function.

We choose a unipolar sigmoid function as the activation function: $f(x) = \dfrac{1}{1+e^{-x}}$ (The assignment interval is determined by the value range of the excitation function.)

The maximum number of training sessions is 11000 and the learning speed is 0.05.

(2) Neuron/hidden neuron input/output expressions

We derived the input/output expressions for the jth neuron based on the weights integrated in task 1 and the results of task 1 and task 2, respectively, as:

Input expression: $\beta_j = \sum\limits_{h=1}^{q} \lambda_{hj} b_h$; Output expression $y_j = f\left(\beta_j + \theta_j\right)$.

The input and output expressions for the hth hidden neuron are

Input expression: $\alpha_h = \sum\limits_{i=1}^{d} c_{ih} x_i$; Output expression $b_h = f\left(\alpha_h + y_h\right)$.

(3) Adjustment of weights

In order to reduce the error signal continuously and make the subsequent result image fit more realistically, we use the gradient descent method to adjust the weights. (The amount of weight adjustment is proportional to the gradient descent of the error)

$$\Delta \omega_{jk} = -\eta \frac{\partial E}{\partial \omega_{jk}} \left(j = 1, 2, ..., m; k = 1, 2, ..., l \right) \tag{10}$$

$$\Delta \upsilon_{ij} = -\eta \frac{\partial E}{\partial \upsilon_{ij}} \left(i = 1, 2, ..., n; j = 1, 2, ..., m \right) \tag{11}$$

(4) obtain an iterative relationship between the implied layer and the input layer, and between the output layer and the implied layer, i.e., continuously adjust the weights $w_{ij}^{\wedge}(N+1)$ from the implied layer to the output layer and $v_{ij}^{\wedge}(N+1)$ from the input layer to the implied layer.

$$\omega_{jk}^{N+1} = \omega_{jk}^{N} + \phi \delta_k^{y} h_j, k = 1, 2, ..., l; j = 0, 1, ..., m$$
$$\upsilon_{ij}^{N+1} = \upsilon_{ij}^{N} + \phi \delta_j^{h} x_i, j = 1, 2, ..., m; i = 0, 1, ..., n \tag{12}$$

(5) compare the global error E and accuracy, if the global error is less than the accuracy of the sample, or the number of iterations exceeds the maximum number, the calculation is over[4].

(6) By taking the values of the above neural network training layers and training times, the 64 indicators bank indicators and bank risk index scores are trained, and then 64 bank indicators are used as the test set as input indicators, and substituted into the neural network prediction model to get the comprehensive score value of each bank[5]. Let the composite score value of each bank be z. The composite score z' is predicted by the neural network and the relative error is calculated as follows.

$$Q = \left(\frac{|z - z'|}{z} \right) * 100\% \tag{13}$$

3. Model establishment and solution

3.1 Input and output data

The input and output data are shown in Table 1:

Table 1: Input and output data

Input Data	Output data	
X2 Total liabilities/total assets	x1 Net profit/total assets	X31(Gross profit + interest)/sales
x3 Working capital/total assets	X4 Current assets/short-term liabilities	x35 Sales profit/total assets
x8 Carrying value of equity/total liabilities	X5[(Cash + short-term securities + receivables - short-term liabilities)/(Operating expenses - depreciation)]*365	x36 Total Sales/Total Assets
X17 Total Assets/Total Liabilities	X6 Total assets/total liabilities	x38 Unchanged capital/total assets
X20 （Inventory*365)/Sales	X7 Profit before tax/total assets	X39 Sales profit/sales
X24 Gross profit (over 3 years)/Total assets	X9 Sales/total assets sales	X40(Current assets - inventories - receivables)/short-term liabilities
X28 Working Capital/Fixed Assets	X10 Equity/Total Assets	X41 Total liabilities/((Profit from operating activities + depreciation)*(12/365))
X29 Log of total assets	X11(Gross profit + extraordinary items + finance costs)/Total assets	X42 Profit from operating activities/sales
X32(Current liabilities*365)/cost of products sold	X12 Gross profit/short-term liabilities	X43 Rotating receivables + inventory turnover (days)
X33 Operating expenses/short-term liabilities	X13(Gross profit + depreciation)/sales	X44(Receivables*365)/Sales
X34 Operating expenses/total liabilities	X14 (Gross profit + interest)/Total assets	X45 Net Income/Inventory
X46(Current assets - inventories)/Short-term liabilities	X15(Total liabilities*365)/(Gross profit + depreciation)	x48(Profit from operating activities - depreciation)/total assets
X47(Inventory*365)/cost of products sold	X16(Gross profit + depreciation)/Total liabilities	x49(Profit from operating activities - depreciation)/sales
X50 Current assets/total liabilities	X18 Gross Profit/Total Assets	X57(Current assets-inventory-short-term liabilities)/(Sales-gross profit-depreciation)
X51 Short-term liabilities/total assets	X19 Gross profit/sales	X59 Long-term liabilities/equity
x52(Short-term liabilities*365)/cost of products sold	x21 Sales(n)/Sales(n-1)	X60 Sales/Inventory
X53 Equity/Fixed Assets	X22 Profit from operating activities/total assets	X63 Sales/short-term liabilities
X55 Working capital	X23 Net profit/sales	X64 Sales/Fixed Assets
X56(Sales-cost of products sold)/sales	X25(Equity-share capital)/Total assets	X30(Total liabilities - cash)/sales
x58 Total Costs/Total Sales	X26(Net profit + depreciation)/Total liabilities	
X62(Short-term liabilities*365)/sales	X27 Profit from operating activities/finance costs	

3.2 Data of five indicators and their weights

The most important five indicators data and their corresponding characteristic roots were obtained as shown in Table 2 :

Table 2: Data of five indicators and their weights

Extraction of principal components	2017	2018	2019	2020	2021
$\lambda 1$	16.656	13.255	11.272	11.356	15.474
$\lambda 2$	6.055	8.206	8.256	8.282	8.684
$\lambda 3$	4.516	5.418	5.147	4.554	5.872
$\lambda 4$	3.692	4.975	3.895	3.871	3.119
$\lambda 5$	3.232	3.458	3.49	3.457	2.903

4. Conclusion

First, the model uses the entropy weight method to select the weight. We guess that we can use the combination of TOPSIS and entropy weight to perform regression analysis to find the optimal and worst weight distribution relationship, and calculate the step size of the remaining weights to these two relationships. , using the fuzzy mathematical idea of (0, 1), the closer it is to 1, the higher the operational efficiency, so as to evaluate the pros and cons of operational efficiency.

Secondly, the PCR method in this paper only selects 5 principal components for analysis, and appropriately increasing the number of selected principal components or introducing factor analysis can more comprehensively explain the internal relationship between the indicators and study the internal mechanism of bankrupt banks.

Thirdly, the artificial neural network has high classification accuracy, strong parallel distributed processing ability, and strong distributed storage and learning ability. In the BP neural network model constructed in this paper, the relationship between the parameters may not be the optimal parameters, so a large number of experimental comparisons are needed to improve the accuracy and fit of the model again.

Finally, the model has a good fitting effect on the whole, but the fitting degree is still relatively low for some banks.

References

[1] Li H.M., Qin G.H.,Hao B., Ji J.Z.. Matching degree calculation method in fuzzy matching[J]. Computer Engineering, 2010,36(06):184-185.
[2] Tian Xiaona,Dong Jing. Fuzzy C-mean clustering for remote sensing image classification[J]. Mine Survey, 2011(03):32-34+36.
[3] Wen Chuanjun,Zhan Yongzhao,Ke Jia. Generalized equilibrium fuzzy C-mean clustering algorithm[J]. Systems Engineering Theory and Practice, 2012,32(12):2751-2755.
[4] Ji G. Stock price trend prediction based on BP neural network[D]. Zhengzhou University, 2018.
[5] Zhang Peng. Comprehensive evaluation research based on principal component analysis [D]. Nanjing University of Science and Technology, 2004.