

Artificial Neural Networks in Default Forecast

József Bozsik

Eötvös Loránd University – Faculty of Informatics
e-mail: bozsik@inf.elte.hu

Abstract

In this article an artificial neural network will be described, which is able to forecast the financial default of a company given enough and good-quality data and well-defined conditions. The method is able to predict the default of a company with the help of financial ratios, which can be calculated from the annual Profit and Loss Statement and Balance Sheet of the company. I tested the method on real companies and real data from 2008. I compared the results with the results of a now-used economical model (discriminance analysis). The comparison shows the reliability of the method and the influence of each parameter to the reliability of the result.

Keywords: neural network, default forecast, finance default, discriminance analysis

1. Introduction

An important scope of artificial neural networks is the forecasting. One of the well-known problem in the financial world is the forecast of defaults. This is an important issue in the current financial situation, especially if we think about the financial crisis. Default models are used to project defaults in the economics. One of the famous models is the so called discriminance analysis. In Hungary the default projections have no long traditions, because the law for bankruptcy and liquidation process¹ was only accepted in 1991. The first default-model [1] was developed by Miklós Virág and Ottó Hajdu in 1996 based on annual report data from 1990 and 1991 using discriminance analysis. From in the survey used 154 company working in processing industry in August 1992, 77 companies were creditworthy and 77 unable to (re)pay. In the model used companies employed at least 300 employees. For the model 17 financial ratios were taken into account. In the next pages I would like to introduce the discriminance analysis, which is the basis of the model.

¹The law Nr. 1991. XLIX. about bankruptcy and liquidation process.

Based on the work of Miklós Virág and Tamás Kristóf I build up and describe in details the solution of the problem with neural networks based on data from 2008. With the analysis of the results I will show under what kind of circumstances and why the neural networks solve the problem better.

2. The model

In order to understand and be able to analyse the solution which is based on a new controlled (supervised) learning method of artificial neuron networks, it is necessary to analyse and introduce the economic solution.

2.1. Discriminance analysis

The discriminance analysis with more variables analyses in the same time the distribution of more ratio and sets up a classification rule, which contains more weighted financial ratio (these are the independent variables of the model) and summarize them in only one discriminant value. [2] The most important criteria of choosing the financial ratios which will be used in the model is that the ratios should have low correlation to each other. Otherwise the added value for the classification will be low. It is worth to begin the construction of the model with a significant ratio and in each further step the less correlated but the second significant ratio should be involved. In the discriminance-function which is a linear combination the values of the financial ratios should be substituted which are calculated from the annual report of the individual companies. In order to be able to classify if a company is able to pay or not, we have to compare the values with the discriminance value.

The general formula of the discriminance function is the following:

$$Z = w_1X_1 + w_2X_2 + \dots + w_nX_n \quad (2.1)$$

In the formula used signs are:

- Z – discriminance value
- w_i – discriminance weights
- X_i – independent variables (financial ratios)
- $i = 1, \dots, n$ where n means the number of financial ratios

The analysis of 2008 as well as the analysis from 1996 showed that the companies which are able to pay and which are not are different in the following financial ratios:

- X_1 – quick liquidity ratio²
- X_2 – cash flow/total debts
- X_3 – current assets³/total assets⁴
- X_4 – cash flow/total assets

²This ratio shows if the company is able to pay immediately.

³Current assets: inventories, receivables, cash and cash equivalents.

⁴Total assets are those assets which are used within one year in the company.

The order of the financial ratios is reflects the discriminance power of the ratios, it means that the most discriminative ratio is the quick liquidity ratio after that the three other ratios. The discriminance function is made by the involvement of these ratios based on the data of 2008 [3]:

$$Z = 1,2387 \cdot X_1 + 1,7153 \cdot X_2 + 2,9761 \cdot X_3 + 0,07158 \cdot X_4 \quad (2.2)$$

The critical Z value is 2,39756, it means if we substitute the values of a company's financial ratios and the result is higher than 2,39756, the company will be classified by the function as solvent (it is able to pay), otherwise insolvent (it is not able to pay).

2.2. The result of the discriminance analysis for the test data

During the test I chose randomly 100 companies from the open database of the webpage of the Hungarian Ministry of Justice and Law Enforcement (<http://www.e-beszamolo.irm.hu/kereses-Default.aspx>). 50 companies were solvent the other 50 insolvent. I applied the data for the introduced discriminance function; the result is shown in the Table 1:

Type of results	The result of the discriminance analysis
Wrong classified companies (pieces)	12
Wrong classified companies which are able to pay (percent)	24%
Wrong classified companies which are not able to pay (pieces)	9
Wrong classified companies which are not able to pay (percent)	18%
Total (pieces)	21
Total (percent)	21%
Classification accuracy (percent)	79%

Table 1: The results of the discriminance analysis based on data of 2008

Our aim is to prepare a default-model which can give more accurate results than the model introduced before. The discriminance analysis gives a base to prepare a default model which uses neural networks.

2.3. The construction of a neural network

According to the international experiences the neural network gives the most reliable forecast results if the learning sample contains 50-50% solvent and insolvent companies (see [4, 5, 6]).

The database chosen from the annual reports of 2008 which I introduced in the last part fulfils all the criteria. In the input layer – as a continual variable – are the financial ratios of the chosen companies. The output layer contains only one neuron, which indicates the ability to pay.

According to the classification the indicator is 0 for the insolvent and 1 for the solvent companies. The default forecast model is prepared by using the mentioned financial ratios, it is shown in the Table 2.

I used a feed-forward layered neural network, where the connections are full between the layers.

There is no connection between the neurons within one layer. I chose this structure because the expected result should be a linear combination of the inputs like in the case of discriminance analysis.

Name of the ratio	Calculation method
Quick liquidity ratio	$\frac{\text{current assets} - \text{inventories}}{\text{short term liabilities}}$
Liquidity ratio	$\frac{\text{current assets}}{\text{short term liabilities}}$
Proportion of Cash (%)	$\frac{\text{cash}}{\text{current assets}}$
Proportion of Cash flow and total debts	$\frac{\text{cash flow}}{\text{total debt}}$
Proportion of current assets (%)	$\frac{\text{current assets}}{\text{total assets}} * 100$
Capital Adequacy ratio	$\frac{\text{fixed assets} + \text{inventories}}{\text{own equity}} * 100$
Turnover of assets	$\frac{\text{total turnover}}{\text{total assets}}$
Turnover of Inventories	$\frac{\text{total turnover}}{\text{inventories}}$
Turnover of Trade Receivables (days)	$\frac{\text{trade receivables} * 360}{\text{total turnover}}$
Indebtedness (%)	$\frac{\text{total debts}}{\text{total assets}} * 100$
Proportion of Own Equity (%)	$\frac{\text{own equity}}{\text{total assets}} * 100$
Solvency ratio	$\frac{\text{total debts}}{\text{own equity}}$
Fixed assets covered by long term loans (%)	$\frac{\text{long term loans}}{\text{fixed assets}} * 100$
Current assets covered by short term loans (%)	$\frac{\text{short term loans}}{\text{current assets}} * 100$
Net Profit Margin (%)	$\frac{\text{net profit}}{\text{total turnover}} * 100$
Return on Equity (%)	$\frac{\text{net profit}}{\text{own equity}} * 100$
Trade Receivables covered by Trade Payables	$\frac{\text{trade receivables}}{\text{trade payables}}$

Table 2: The used financial ratios and the way of calculation

There is no need for feedback and connections in-between the layers. The neurons which are in the network have homogeny framework. The function of every inside neuron is:

$$f(x) := \frac{1}{1 + e^{-x}} \quad (2.3)$$

It was favourable to choose the logistic activation function for the construction of the network, because it can be differentiated in every point, it means it is smooth enough for the description of a system with continuous variables.

I used the Backpropagation model (see [7]) for the teaching. Let me introduce the following indicators:

- Let the input layer be indicated by 0 . layer for the easier fitting
- Let me indicate the l -layer's i -ranked neuron's input value as: $a_0^{l-1}, \dots, a_{n_i}^{l-1}$
- Let me indicate the l -layer's i -ranked neuron's input weights as: $w_0^{l,i}, \dots, w_{n_i}^{l,i}$
- Let me indicate the l -layer's i -ranked neuron's output values as: a_i^l

The Figure 1 shows the structure which refers to the indicators introduced above:

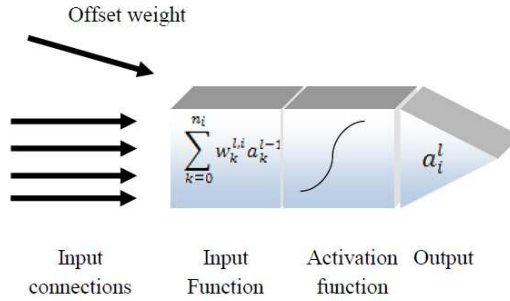


Figure 1: The l -layer's i -ranked neuron

Accordingly the steps of the error backpropagation algorithm which is used for the teaching are following (see [8]):

1. Calculate each neuron's output from the $x = (a_0^0, \dots, a_{n_0}^0)$ input vector by layers: a_i^l , after the calculation we receive the output of the output layers which i indicate by: a_{out} .
2. Calculate the errors for the output values:
 $e_{output} = a_{out} (1 - a_{out}) (a_{expected} - a_{out})$ and the changes of the weights.
3. Calculate the error of the neurons by layers from back to the forth:

$$e_i^l = a_i^l (1 - a_i^l) \sum_{k=1}^{n^{l+1}} e_k^{l+1} w_k^{l+1,i} \quad (2.4)$$

and the changes of the weights where η means the learning-coefficient:

$$\Delta w_i^{l,j} = \eta e_i^l a_i^{l-1} \quad (2.5)$$

4. Modify the weights of the network:

$$w_i^{l,j} = w_i^{l,j} + \Delta w_i^{l,j} \quad (2.6)$$

Let these steps repeat until the input layer will be reached. With this step the weight-modifications will be validated for the whole network, it means the correction referring to the error is done for every layer and every neuron in the network.

3. The results of the neural network

In the first step of the construction of the neuron network model I divided the total sample (the database with 100 company data) to teaching and testing samples. In the experiment 75% of the total sample is assigned as teaching sample and the rest 25% is assigned to testing sample. The teaching and the testing samples are randomly assigned. In each step of the teaching process the teaching patterns are also randomly chosen for the network input.

The strategy for teaching the neural network – it means to reallocate the weights – is the following:

1. by certain number of running the teaching cycle
2. by running until a certain accuracy is met (until square error limit)

If we choose the square error limit, we have to define the number of teaching cycles, because it is not sure, that the given accuracy can be met, or in case of wrong chosen error limit the network can be over-learned. This happens often in case of wrong calibrated networks, but by testing and measuring the level of teaching (with test patterns) can be set up to almost optimal. The over-learning problem is shown on the Figure 2.

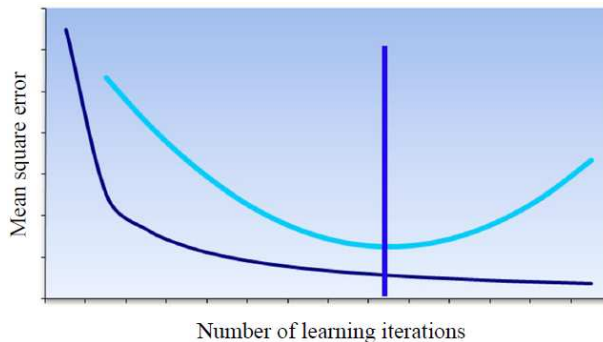


Figure 2: The outcome of the error of the teaching and testing patterns during the learning cycle

The over-learning problem can be avoided, if we monitor the error of the teaching and the testing patterns parallel. It should be allowed to iterate with teaching patterns only until the error of the teaching and the testing patterns are close to each other. Naturally in case of different topological networks the number of needed iterations can be very different. This is shown in the Table 3. In case of increasing the number of the teaching patterns for each topology, the error of the teaching pattern decreased, but on the other hand we receive higher errors for the testing patterns.

Name	Number of neurons in two intermediate layers (pieces)					
	4-4	5-4	6-4	4-5	5-5	6-5
Number of teaching cycles	400	400	700	700	800	800
Error of teaching patterns (%)	22,7	24	22,7	26,7	21,3	17,3
Error of testing patterns (%)	16	20	16	16	12	8

Table 3: The teaching cycle of six neural network

Using three or less intermediate level neural networks – independent from the number of neurons – it did not succeed to reach small enough square error during the test experiments. This suggests that the network with this construction is not able to reach the appropriate accuracy, which shows the network's poor ability of estimation. This outcome is not surprising, because the number of the input values in the network is 17, which describes a more complex system, so accordingly the network with few hidden layers is not able to project. This is the reason why I continued my experiments with a neural network with at least four hidden layers.

During the experiments it became also clear, that it is not worth to use less than four neurons in one intermediate layer, because in this case the level of average square error high as well. This attribute of the network refers to that few neurons in the intermediate levels cannot give reliable results for the forecast.

The model's ability to forecast will worsen, if

- there are seven or more neurons in the first intermediate layer
- there are six of more neurons in the second intermediate layer

The best result which helps the projections was given, when there were 4-6 neurons in the first intermediate layer and 4-5 neurons in the second intermediate layer. In this case the average square error of the teaching patterns was between 17,3% and 26,7%, while the average square error of the testing pattern was between 8% and 20% (See Table 3).

Beside the average square error there is one important characteristic of the forecast system, this is the so called classification accuracy. According to this accuracy ratio the default forecast models can be classified. The Table 4 and 5 shows the classification accuracy of the teaching and testing patterns.

Types of classification in teaching patterns	Number of neurons in two intermediate layers (pieces)					
	4-4	5-4	6-4	4-5	5-5	6-5
Wrong classified solvent (pieces)	6	3	8	3	6	5
Wrong classified insolvent (pieces)	11	15	9	17	10	8
Total wrong classified (pieces)	17	18	17	20	16	13
Total wrong classified (percent)	22,7	24	22,7	26,7	21,3	17,3
Classification Accuracy (percent)	77,3	76	77,3	73,3	78,7	82,7

Table 4: The classification accuracy in teaching patterns

Types of classification in testing patterns	Number of neurons in two intermediate layers (pieces)					
	4-4	5-4	6-4	4-5	5-5	6-5
Wrong classified solvent (pieces)	2	4	1	3	2	1
Wrong classified insolvent (pieces)	2	1	3	1	1	1
Total wrong classified (pieces)	4	5	4	4	3	2
Total wrong classified (percent)	16	20	16	16	12	8
Classification Accuracy (percent)	84	80	84	84	88	92

Table 5: The classification accuracy in testing patterns

4. Summary

The result of the experiment shows us, that the highest classification accuracy was met from the neural networks trained by the 100 company data, which have two hidden layers (in the first hidden layer contains six, in the second hidden layer five neurons).

This result is in accordance with the result of Miklós Virág and Tamás Kristóf (see [6]), where also the similar constructed network was the most successful.

In the experiment the 5-6 network reached 85% classification accuracy, which is shown in the Table 6.

Taking into consideration the classification accuracies on the examined networks this topology seems to be the most successful, but it should not be forgotten that the ability of estimation was trained and tested on a relative small sample database.

Comparing our results with the results of the discriminance analysis the 6-5 network shows better results, but we have to take into consideration the important fact that the discriminance function was set up from 500 randomly chosen companies by the authors. This remark is very important, because this network was only trained and tested on 100 company data. Despite this fact the network could achieve good results compared to the traditional method. But as for the more variables mathematic models, like the statistical models and the neural network based model, it is suggested to examine the problem with more methods, and if they all give similar results, only then should it be accepted.

There is a surprising conclusion out of the results of the experiments: all the six models had more first-type error, than second type. The first-type error happens, if the model classifies the insolvent companies into solvent companies incorrectly, the second-type error happens if the network classifies the solvent companies into insolvent companies. This suggests that all the six neural network based model classifies incorrectly insolvent companies to solvent companies.

Name	Number of neurons in two intermediate layers (pieces)					
	4-4	5-4	6-4	4-5	5-5	6-5
Wrong classified solvent (pieces)	8	7	9	6	8	6
Wrong classified insolvent (pieces)	13	16	12	18	11	9
Total wrong classified (pieces)	21	23	21	24	19	15
Classification accuracy (percent)	79	77	79	76	81	85

Table 6: The classification accuracy of the six neural network model

References

- [1] VIRÁG MIKLÓS, HAJDU OTTÓ: Pénzügyi mutatószámokon alapuló csődmódelleszámítások. *Bankszemle*, XV. évf. 5. sz. (1996), 42–53.
- [2] KOVÁCS ERZSÉBET: Pénzügyi adatok statisztikai elemzése, Egyetemi Tankönyv, *Budapesti Corvinus Egyetem Pénzügyi és Számviteli Intézet*, (2006)
- [3] KOVÁCS GÉZA: Diszkriminanciaanalízis 2008. évi adatok alapján, *Online Study*: <http://kozgazdasag.uw.hu/kovacsg/dizskrim2007/dizskrim2007.html> (2009.09.10.)
- [4] ODOM, M.D., SHARDA, R.: A Neural Network Model For Bankruptcy Prediction. *Proceeding of the International Joint Conference on Neural Networks*, San Diego, június 17–21. Volume II. IEEE Neural Networks Council, Ann Arbor, (1990), 163–171.
- [5] TAM, K. – KIANG, M.: Managerial applications of the neural networks: The case of bank failure predictions. *Management Science*, Vol. 38. No. 7. (1992), 416–430.
- [6] VIRÁG MIKLÓS – KRISTÓF TAMÁS: Az első hazai csődmódelleszámítás neurális hálók segítségével. *Közgazdasági Szemle*, LII. évf., (2005), 144–162.
- [7] WERBOS, P.: Beyond regression: new tools for prediction and analysis in the behavioural sciences. Ph.D Thesis, Harvard University, Cambridge M.A. (1974)
- [8] GREGORICS TIBOR: Mesterséges intelligencia alapjai, *Electronic version of a university lecture*: <http://people.inf.elte.hu/gt/mi/neuron/neuron.pdf> (2009.09.15.)