

The Influence of Hidden Neurons Factor on Neural Network Training Quality Assurance

Peter Grabusts, Aleksejs Zorins

Rezekne Higher Educational Institution, Faculty of Engineering,
Atbrivosanas alley 115, Rezekne, LV-4601, Latvia

Abstract. The work shows the role of hidden neurons in the multilayer feed-forward neural networks. The numeric expression of hidden neurons is usually determined in each case empirically. The methodology for determining the number of hidden neurons are described. The neural network based approach is analyzed using a multilayer feed-forward network with backpropagation learning algorithm. We have presented neural network implementation possibility in bankruptcy prediction (the experiments have been performed in the Matlab environment). On the base of bankruptcy data analysis the effect of hidden neurons to specific neural network training quality is shown. The conformity of theoretical hidden neurons to practical solutions was carried out.

Keywords: bankruptcy prediction, financial ratio, hidden neurons, neural networks, backpropagation.

I INTRODUCTION

In literature on artificial neural networks there is often found the concept - "hidden neuron" or "hidden layer". It is generally said that for the network training with complex input data the hidden neurons are required. Indeed, the role of hidden neurons is often not clear. Also, in special literature, their role and functions are described minimally. For example, such types of definitions are given: "Between these layers there is one or more interlayers (hidden layers), which perform the knowledge accumulator functions" [2].

If formally the neural network is treated as consisting of the input layer and output layer, then in the hidden layer a particular training algorithm operation is actually realized.

The work gives an attempt to explain the role of hidden neurons, explain the situation when there is a need for the hidden neurons, the methods are described that allow to select the optimal number of hidden neurons, and also to know whether dealing with specific input data - in the training process the hidden neurons will be needed or not. In practical part on the basis of bankruptcy data example the selection of the number of hidden neurons for training quality assurance is described experimentally.

II THE EMPIRICAL DETERMINATION OF THE NUMBER OF HIDDEN NEURONS

Neural network, in general, has the following structure:

- The network consists of many interconnected neurons;
- Neurons are arranged in levels or layers;
- The first layer introduces the data into the neural network;
- The output determines the response of the network to the input data;
- Between the first and the last layer there may be one or more hidden layers which performance the accumulator functions in the network.

In the work [13] the author J. Kinser describes the methodology that seems interesting as it offers to determine the need for the hidden neurons before training by analyzing the input data. It is called *hidden neuron* theory and describes the procedure which extracts all the necessary information to determine the network architecture before the training process from the input data. The number of possible hidden neurons, its specific role and position for all training sets of data pairs can be determined. The method allows finding network architecture with a minimal number of hidden neurons.

The idea is as follows. The number of hidden neurons depends directly on the internal structure of training data and its essence is to resolve conflicts in the input data - if the input data contains conflicts, then the neural network needs hidden neurons.

It can be explained with XOR example. In XOR problem there are 4 training elements. As the activation function the threshold function is used:

$$y(a) = \begin{cases} 1, & a > \gamma \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $0 < \gamma < 0$.

For training data pairs (x:y) the following inequalities must be:

$$\sum_i w_{ji}x_i \begin{cases} > \gamma, & y = 1 \\ < \gamma, & y = 0 \end{cases} \quad (2)$$

Thus, XOR problem is converted with a number of inequalities:

$$\begin{aligned} 0 + 0 &< \gamma \\ 0 + w_{12} &\geq \gamma \\ w_{11} + 0 &\gg \gamma \\ w_{11} + w_{12} &< \gamma \end{aligned} \quad (3)$$

It is clearly seen that, if $W_{12} \geq \gamma$ and $W_{11} \geq \gamma$ is fulfilled, then inequality $W_{11} + W_{12} < \gamma$ can not be fulfilled. It is said that inequalities are in a conflict. The solution of the conflict is introduction of hidden neurons, as it is shown in Figure 1.

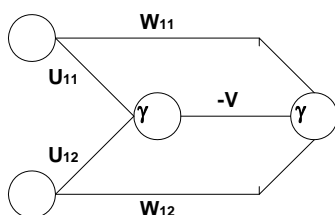


Fig.1. Example solution of XOR problem

The corresponding inequality systems will be as follows:

$$\begin{aligned} 0 + 0 &< \gamma & 0 + 0 + 0 &< \gamma \\ 0 + U_{12} &< \gamma & 0 + W_{12} + 0 &\geq \gamma \\ U_{11} + 0 &< \gamma & W_{11} + 0 + 0 &\gg \gamma \\ U_{11} + U_{12} &\geq \gamma & W_{11} + W_{12} + V &< \gamma \end{aligned} \quad (4)$$

This means that the conflict has been resolved through the introduction of additional neurons. Searching the conflicts in large training sets the weak point of this method, unfortunately, is in solving the inequality systems.

III PRACTICAL RECOMMENDATIONS FOR THE SELECTION OF THE NUMBER OF HIDDEN NEURONS

Different authors have given different evaluations of the hidden neurons. The most common evaluation is as follows: if there are N input elements, then the total number of input vectors is 2^N . The maximum number of hidden neurons is 2^{N-1} .

The authors of this work [9] offer the following evaluation (known as Baum – Haussler rule):

$$N_{hidden} \leq \frac{N_{train}E_{tolerance}}{N_{pts}+N_{output}} \quad (5)$$

where:

- N_{hidden} – number of hidden neurons;
- N_{train} – number of training elements;
- N_{pts} – number of learning elements;
- N_{output} – number of output neurons;
- $E_{tolerance}$ – error.

Several authors, among them [1], [11], give the following, seemingly simple, practical recommendations for the feed-forward network implementation:

- Use one hidden layer;
- Use fewer hidden neurons, if possible.

How many hidden layers?

There is no reason to use more than one hidden layer – otherwise the network training can take a long time. Of course, there is always a possibility that several layers perfectly solve the complex problem. Therefore, in the feed-forward model practical realization for the first time it is recommended to use only one hidden layer. If a large number of hidden neurons in the first layer do not offer a good solution to the problem, it is worth trying to use a second hidden layer, reducing the total number of hidden neurons.

How many hidden neurons?

The need to choose the right number of hidden neurons is essential. Using too little – the network lacks the resources for training algorithm implementation. Using too much – increases the training time, even it can be impossible to be adequately trained within a reasonable period of time. Therefore it is necessary to use an absolute minimum of hidden neurons that solves the problem.

For most of the tasks the *geometric pyramid rule* can be used. It states that the number of neurons should be reduced from the input to the output, as it is seen in Figure 2.

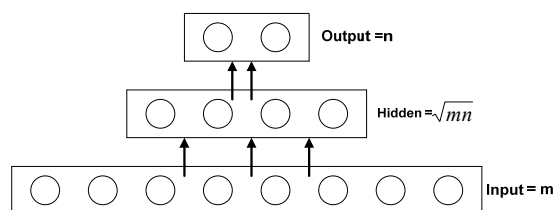


Fig.2. Typical three-layer network 8-4-2

Similarly, the four-layer network could be set up, such as 8-4-2-1. The number of hidden neurons in layers HID_1 and HID_2 should be selected according to the following relationship:

$$HID_1 = nr^2 \quad (6)$$

$$HID_2 = nr \quad (7)$$

where $r = \sqrt[3]{\frac{m}{n}}$.

The above written formulas give only a rough evaluation. In practice, for each specific task the required number of hidden neurons is determined experimentally.

IV EXPERIMENTAL PART – ANALYSING BANKRUPTCY DATA

A. Bankruptcy financial ratios

Bankruptcy diagnostics is a directed financial analysis system which area is a crisis situation control at the enterprises. For this purpose, in the analysis of the general financial situation of the company a separate group of financial ratios is made, using which it is possible to reason about the threat of bankruptcy. In general case there is no theoretical background as to which financial ratios might be used in different bankruptcy studying models. A lot of researchers performing an analysis of the bankrupt enterprise proceed in this way: they calculate several ratios and then select potentially most significant of them.

Beaver [10], Altman [5], and Ohlson [14] are considered the pioneers of the empirical approach, Beaver was one of the first who has applied balance sheet data in bankruptcy research. His analysis was comparatively simple and was based on studying one financial ratio and comparing it with other ratios. He has concluded that ratio R4: Cash flow / Total debt is a very essential indicator which has to be accounted in bankruptcy analysis. Beaver’s works became a beginning of multicriteria analysis application, which was later developed by Altman et al.

Altman employed classical multivariate discriminant analysis (MDA) in his research, Altman’s Z-scores [6] are widely used as input data in neural network algorithms. It was noted that Z-score has an essential drawback – actually it can only be related to large companies whose stocks are quoted in the stock exchange.

B. Neural network approaches

Neural network approach to bankruptcy prediction [7], [8] became extremely popular in the nineties of the last century. Starting from 1990 research on neural network application in bankrupt analysis was performed periodically: Odom and Sharda [4], Tam and Kiang [17] et al.

Odom and Sharda were one of those who first employed neural network techniques in bankruptcy analysis. In the input of the network, Altman’s Z-scores about 128 companies were used. It was shown that neural network approach yields better results than MDA.

Tam and Kiang have compared different techniques applied in bankruptcy diagnostics (MDA, single layer network and multilayer network) and have shown that

in the „one-year-ahead” data the multilayer network was most effective whereas in the “two-year-ahead” data the other method turned to be most effective.

Analysing the neural network application in bankruptcy analysis, these conclusions were made:

- Neural networks ensure approximately 90% accuracy compared to the 80-85% accuracy of other methods;
- Bankruptcy can be predicted several years before it happens, the accuracy of prediction being practically the same for the “one-year-ahead” data and for the “two-year-ahead”.

Formally speaking, bankruptcy prediction task practically does not differ from that of classical pattern recognition. Financial ratios are forwarded to the network input and network learning is accomplished. Use of hidden neurons is made. In the network output there are only two states: bankrupt or non-bankrupt. General scheme of the network that will be used in the experiments presented here, is shown in Figure 3.

For a detailed description of this well known type of network see [3] and [16].

C. Experimental data

The data on firm bankruptcy were taken from [15], [18]. For the purpose of experiments, balance sheet data of 63 companies were used - 46 bankruptcy and 17 not bankruptcy (for other records from [15] information was incomplete for calculating the necessary financial ratios). It was decided to calculate the following financial ratios on the basis of the data available and further use them in all the experiments:

- R2: Cash Flow / Current Liabilities;
- R3: Cash Flow / Total Assets;
- R7: Current Assets / Current Liabilities;
- R9: Current Assets / Total Assets;
- R31: Working capital / Total assets.

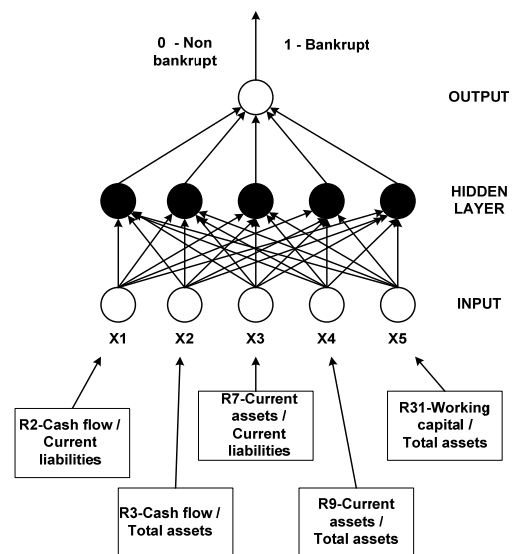


Fig.3. Multilayer feed-forward neural network architecture

The multilayer feed-forward neural network has been chosen as a neural network model. A series of experiments have been performed in order to choose the number of hidden neurons suitable for the purpose of learning. From now on, all the experiments are based on the 5-5-1 model. The parameters of the chosen network model were as follows:

- Input nodes - 5 neurons;
- Hidden layer - 5 neurons;
- Output nodes - 1 neuron (1- bankrupt, 0 - not bankrupt);
- Learning rate - 0,25;
- Stopping condition - the training is stopped if MSE=0,5;
- Momentum rate - α ;
- Slope of the tanh activation function - β .

The input nodes represent the financial ratios that the model will use to predict bankruptcy. The learning and momentum rates define the rate and accuracy with which the backpropagation algorithm converges on the error minimization solution. Larger numbers for momentum are used to achieve faster convergence, but run the risk of skipping over the optimal solution. The final parameters were obtained after extensive experiments.

In the first part of the experiment, the learning quality dependence on α value was checked. The given experimental model has shown that at the α values within 0,8 the following occurs: the algorithm does not converge or else correct network learning does not happen. Valid learning results were obtained at $\alpha=0,8$ and $\alpha=0,9$ [12].

In the second part of the experiment the learning quality dependence on the slope of the *tanh* activation function - $\beta \in [0,1; 1]$ was checked. As a result of experiments, it was found that acceptable learning quality was achieved at specific β values only. Table 1 shows the most significant results and misclassified cases [12].

TABLE 1.

EXPERIMENTAL RESULTS (PARAMETERS α , β AND ITS MISCLASSIFIED CASES)

PARAMETER α	PARAMETER β	EPOCHS	NO OF MISCLASSIFIED CASES	CASES
$\alpha = 0,8$	$\beta = 0,8$	41	9	14,26,36,37,41,58,59,60,62
	$\beta = 0,9$	889	6	37,50,58,59,60,62
$\alpha = 0,9$	$\beta = 0,8$	46	9	14,26,35,36,37,41,58,59,62
	$\beta = 0,9$	1489	7	37,50,58,59,60,62,63

In Figure 4 error graph for $\alpha=0,8$ and $\beta = 0,9$ are shown.

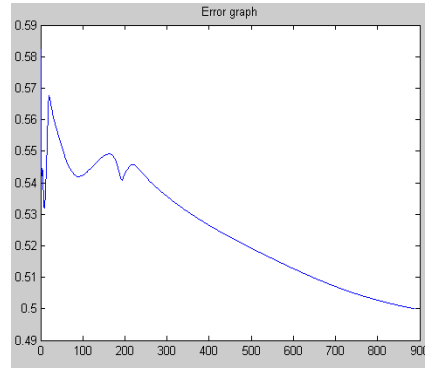


Fig.4. Error graph for $\alpha=0,8$ and $\beta = 0,9$

All the results on the misclassified cases of the methods employed are summarised in Table 2.

From the data of Table 2 it can be seen that for the specific bankruptcy data sample, all the methods are unable to classify data vectors 37, 38, 59 and 62. Calculating in absolute numbers, we obtain that NN-2 correctly classified 88,9% cases, the potential function method and NN-4 - 88% cases, NN-1 and NN-3 - 85,7%. It can be concluded that for the given data sample the neural network method performs bankruptcy data classification more effectively, which actually corresponds to the conclusions about the results achieved by Tam and Kiang [17].

TABLE 2.

SUMMARY TABLE ABOUT USED METHODS AND MISCLASSIFIED CASES

METHOD	NN-1	NN-2	NN-3	NN-4
PARAMETERS	$\alpha = 0,8$	$\alpha = 0,8$	$\alpha = 0,9$	$\alpha = 0,9$
	$\beta = 0,8$	$\beta = 0,9$	$\beta = 0,8$	$\beta = 0,9$
MISCLASSIFIED CASES	14		14	
	26		26	
			35	
	36		36	
	37	37	37	37
	41		42	
		50		50
	58	58	58	58
	59	59	59	59
	60	60		60
62	62	62	62	
			63	

D. The effect of hidden neurons on experimental data

Since the network best option was with $\alpha=0,8$ and $\beta=0,9$, an experimental research was carried out – how the selecting the number of hidden neurons affects the overall quality of the network and whether it corresponds to theoretical principles of selecting the number of hidden neurons. In Table 3 the performance

of neuron network at different hidden neurons values is shown (see Table 3 and Figure 5).

TABLE 3.
NUMBER OF HIDDEN NEURONS AND QUALITY EVALUATION

METHOD	NN-1	NN-2	NN-3	NN-4	HIDDEN NEURONS
PARAMETERS	$\alpha = 0,8$ $\beta = 0,8$	$\alpha = 0,8$ $\beta = 0,9$	$\alpha = 0,9$ $\beta = 0,8$	$\alpha = 0,9$ $\beta = 0,9$	
QUALITY (%)	88,9 88,9 88,9	88,9 88,9 88,9	87,3 87,3 87,3	90,5 88,9 88,9	1 2 3
	85,7	88,9	84,1	88,9	4
	82,5	88,9	84,1	88,9	5
	82,5	88,9	84,1	88,9	6
	85,7	88,9	82,5	88,9	7
	82,5	88,9	85,7	88,9	8
	82,5	88,9	85,7	88,9	9
	85,7	88,9	84,1	88,9	10
	87,3	88,9	87,3	88,9	11
	87,3	88,9	82,5	88,9	12

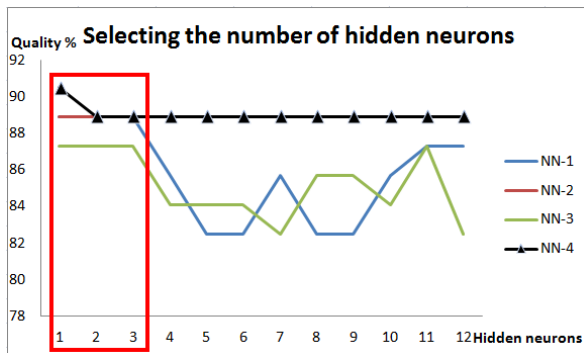


Fig.5. Selecting the number of hidden neurons for bankruptcy data analysis

It can be concluded, that for high-quality network training in this particular case it is enough to use as minimum 1 neuron in the hidden layer, in order to ensure an acceptable quality of training that also corresponds to the theoretical evaluation: according to the formula (6) we have:

$$HID_1 = nr^2 = \left(\sqrt[3]{\frac{m}{n}} \right)^2 = 1,7^2 = 2,9 \approx 3$$

that is maximum 3 neurons in the hidden layer, which correspond to the theoretical evaluation. Further

increase of the hidden neurons, as it is shown in Table 3, is not needed.

V CONCLUSIONS

Working with feed-forward network architecture models and using the back propagation learning algorithm, we face the problem to determine the number of hidden neurons. This paper presents some methods of determining number of hidden neurons. The last chapter gives practical recommendations for testing specific data samples which analyze neural network input data and test the situations where hidden neurons are necessary. This could make it possible to identify the number of hidden neurons before the network training to build up the optimal network. The method is suitable only for feed-forward neural network model. In the experimental part, the application of bankruptcy data leads to the conclusion that using 3 neurons in the hidden layer in a specific case is entirely sufficient for providing the neural network training quality. It allows saving time in network training and testing.

VI REFERENCES

- [1] T. Masters, *Practical Neural Network recipes in C++*. Academic Press, 1993.
- [2] I. Alexander and H. Morton, *An Introduction to Neural Computing*. Chapman & Hall, London, 1991.
- [3] J. Hertz, A. Krogh and R.G. Palmer, *Introduction to the theory of neural computation*. Addison Wesley, 1991.
- [4] M. Odom and R. Sharda R, A neural network model for bankruptcy prediction, In Proc. Int. Joint Conf, Neural Networks, San Diego, CA, 1990.
- [5] E. Altman E, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *Journal of Finance*, vol. 13, pp.589-609, 1968.
- [6] E. Altman, R. Haldeman and P. Narayanan, "ZETA analysis. A new model to identify bankruptcy risk of corporations," *Journal of Banking and Finance* 1, pp. 29-54, 1977.
- [7] A. Atiya A, "Bankruptcy prediction for credit risk using neural networks: A survey and new results," *IEEE Transactions on Neural Networks*, Vol. 12, No. 4, pp. 929-935, 2001.
- [8] B. Back, T. Laitinen and K. Sere, "Neural networks and bankruptcy prediction: funds flow, accrual ratios and accounting data," *Advances in Accounting* 14, pp. 23-37, 1996.
- [9] E.B. Baum and D. Haussler, "What size net gives valid generalization," *Neural Computation*, 1, pp. 151-160, 1988.
- [10] R. Beaver, "Financial ratios as predictors of failure. Empirical Research in Accounting: Selected Studies," *Accounting Research*, vol. 4, pp.71-111, 1966.
- [11] K. Gnana Sheela and S.N. Deepa, "Review on Methods to Fix Number of Hidden Neurons in Neural Networks," *Mathematical Problems in Engineering*, Volume 2013.
- [12] P.Grabusts, "Analysing Bankruptcy Data with Neural Networks," 10th International Conference on Soft Computing, Mendel2004, Brno, Czech Republic, June 16.-18., pp.111-117, 2004.
- [13] J.M. Kinser J.M., „The determination of Hidden Neurons,” *Optical Memories and Neural Networks*, 5(4), 245-262, 1996.
- [14] J. Ohlson J, "Financial ratios and the probabilistic prediction of bankruptcy," *Accounting Res*, vol. 18, pp. 109-131, 1980.
- [15] G. Rudorfer G, "Early bankruptcy detecting using neural networks," *APL Quote Quad, ACM New York*, vol. 25, N. 4, pp. 171-178, 1995.

- [16] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning representations by backpropagating errors," *Nature*, 323(9):533-536, 1986.
- [17] K. Tam and M. Kiang, "Managerial applications of the neural networks: The case of bank failure predictions," *Management Science*, vol. 38, pp. 416-430, 1992.
- "Bankruptcy data," [Online]. Available: <http://godefroy.sdf-eu.org/ap195/ratios95.zip> [Accessed: March 30, 2015].