Asian Journal of Information Technology 6 (3): 329-341, 2007

Evaluating Financial Sector Firm's Creditworthiness for South-Asian Countries

Syed Adnan Haider Ali Shah Bukhari
Department of Computer Science and I.T, Faculty of Computer Science,
Federal Urdu University of Arts, Science and Technology,
Gulshan-e-Iqbal Campus, Karachi, Pakistan

Abstract: Can financial sector firms learn to be rational in their business activities? The answer depends on the institutionally bounded constraints to learning. From an evolutionary perspective the functionality of learning to be rational creates strong incentives for such learning without, however, guaranteeing that each member of the particular economic species actually achieves increased fitness. We investigate this issue for a particular economic species, namely, commercial banks. The purpose of this study is to present two main contributions. First we review the topic of creditworthiness of financial sector firms, with emphasis on Neural-Network (NN) models. Second, we provide an empirical analysis by using an NN model with back propagation mechanism for creditworthiness decision. The data is taken from three of the South Asian countries; Bangladesh, India and Pakistan. Our empirical findings were provided for three NN architectures by applying training and testing samples constructed from data of the firms that applied for credit in regional banks of South Asian countries for the period 1999 - 2005. To study the effect of proportion between the number of firms that obtained and did not obtain credit, three proportions of the training and testing set compositions were created: A, B, C, finally, the empirical methodology used in our analysis evaluates the classification accuracy in terms of errors made by the NN.

Key words: Artificial neural networks, back propagation mechanism, regional banks, creditworthiness

INTRODUCTION

Creditworthiness prediction of financial sector firms has long been an important and widely studied topic. The main impact of such research is in bank lending. Banks need to predict the possibility of default of a potential counterparty before they extend a loan. This can lead to sounder lending decisions and therefore result in significant savings. The aim of this research was to use the back propagation algorithm to simulate a bank's decisions on giving credit. Presented here are several network architectures constructed and trained during a certain number of iterations for randomly chosen weights. Numerical experiments were provided by applying the computer program Neural-Works Professional. The results of the experiments were evaluated in terms of errors and they compare the decisions generated by the network to the actual decisions made by the bank.

To get an idea about the potential impact of the creditworthiness prediction problem, we note that the volume of outstanding debt to corporations in the United States is about $5 trillion. An improvement in default prediction accuracy of just a few percentage points can lead to savings of tens of billions of dollars. In addition to avoiding potentially troubled obligors, the research can also benefit in other ways. It can help in estimating a fair value of the interest rate of a loan (that reflects the creditworthiness of the counterparty). It can help in accurately assessing the *credit risk* of bank loan portfolios.

The credit risk problem is essentially the computation of the *loss level*, which is defined as the level for which there is a probability of 1% that the loss incurred in the portfolio will exceed that level in a particular time period. Credit risk has been the subject of much research activity, especially after realizing its practical necessity after a number of high profile bank failures in Asian countries. As a result, the regulators are acknowledging the need and are urging the banks to utilize cutting edge technology to assess the credit risk in their portfolios. Measuring the credit risk accurately also allows banks to engineer future lending transactions, so as to achieve targeted return/risk characteristics. The other benefit of the prediction of bankruptcies is for accounting firms. If an accounting firm audits a potentially troubled firm and misses giving a warning signal (say a "going concern" opinion), then it faces costly lawsuits.

The traditional approach for banks for credit risk assessment is to produce an internal rating, which takes into account various quantitative as well as subjective factors, such as leverage, earnings, reputation, etc., through a scoring system. The problem with this approach is of course the subjective aspect of the prediction, which makes it difficult to make consistent estimates. Some banks, especially smaller ones, use the ratings issued by the standard credit rating agencies. The problem with these ratings is that they tend to be reactive rather than predictive (for the agencies to change a rating of a debt, they usually wait until they have a considerably high confidence/evidence to support their decision). There is a need, therefore, to develop fairly accurate quantitative prediction models that can serve as very early warning signals for counterparty defaults.

There are two main approaches to loan creditworthiness prediction. The first approach, the *structural approach*, is based on modeling the underlying dynamics of interest rates and firm characteristics and deriving the default probability based on these dynamics. The second approach is the *empirical* or the *statistical approach*. Instead of modeling the relationship of default with the characteristics of a firm, this relationship is learned from the data. The focus of this article is on the empirical approach, especially the use of NNs.

REVIEW OF RELEVANT LITERATURE

The pioneers of the empirical approach are Beaver, Altman and Ohlson. Beaver (1966) was one of the first researchers to study the prediction of creditworthiness using financial statement data. However, his analysis is very simple in that it is based on studying one financial ratio at a time and on developing a cutoff threshold for each ratio. The approaches by Altman and Ohlson are essentially linear models that classify between healthy/bankrupt firms using financial ratios as inputs. Altman (1968) uses the classical Multivariate Discriminant Analysis technique (MDA). It is based on applying the Bayes classification procedure, under the assumption that the two classes have Gaussian distributions with equal covariance matrices. The covariance matrix and the class means are estimated from the training set. Altman used the following financial ratios as inputs:

- Working capital/total assets;
- Retained earnings/total assets;
- Earnings before interest and taxes/total assets;
- Market capitalization/total debt;
- Sales/total assets.

These particular financial ratios have been widely used as inputs, even for NNs and other nonlinear models. They are described in more detail in the next subsection. Ohlson (1980) introduced the Logistic Regression approach (LR) to the creditworthiness prediction problem. It is essentially a linear model with a sigmoid function at the output (it is thus similar to a single-neuron network). Because the output is in between 0 and 1, the model has a nice probabilistic interpretation. Ohlson used a novel set of financial ratios as inputs. Both the MDA model and the LR model have been widely used in practice and in many academic studies. They have been standard benchmarks for the loan default prediction problem.

Research studies on using NNs for creditworthiness prediction started in 1990 and are still active now. There are a number of reasons why a nonlinear approach would be superior to a linear approach. It can be argued that there are saturation effects in the relationships between the financial ratios and the prediction of default. For example, if the earnings/total assets changes say by an amount of 0.2, from 0.1 to 0.1, it would have a far larger effect (on the prediction of default) than it would if that ratio changes from say 1.0 to 1.2. One can also argue that there are multiplicative factors as well. For example, the potential for default for a firm with negative cash flow gets more amplified if it has large liabilities. The reason is that highly leveraged firms have a harder time borrowing money to finance their deficits. As will be seen from the review below, NNs have generally outperformed the other existing methods. Currently, several of the major commercial loan default prediction products are based on NNs. For example, Moody's *Public Firm Risk Model* is based on NNs as the main technology. Many banks have also developed and are using proprietary NN default prediction models.

There has been also a number of other review papers. For example, Vellido *et al.* (1999) survey the use of NNs in business applications. This survey includes a section on creditworthiness prediction. Also, the survey of Wong *et al.* (1997) on NNs in business applications includes some references on the creditworthiness prediction problem. Dimitras *et al.* (1996) provide a survey on the classical empirical approaches. Zhang *et al.* (1999) include in their paper a nice review of existing work on NN creditworthiness prediction. The majority of the NN approaches to default prediction use multilayer networks. Since this is the dominant approach, henceforth when we mention NNs we mean multilayer networks.

One of the first studies to apply NNs to the creditworthiness prediction problem was the work by Odom and Sharda (1990). Odom and Sharda used Altman's financial ratios (described above) as inputs to the NN and applied their method, as well as MDA as a comparison, to a number of bankrupt and solvent US firms, where the data used for the bankrupt firms are from the last financial statement before declaring creditworthiness. They considered 128 firms and performed several experiments where they varied the proportion of bankrupt/healthy firms in the training set. The NNachieved a Type I correct classification accuracy in the range of 77.8 to 81.5% (depending on the training setup) and a Type II accuracy in the range of 78.6 to 85.7%. The corresponding results for MDA were in the range of 59.3 to 70.4% for Type I acuracy and in the range of 78.6 to 85.7% for Type II acuracy. Let us now discuss why the particular indicators of (which are the same as Altman's indicators) have been chosen. Most other studies use indicators similar in nature and the analysis presented will somewhat apply to these studies as well. A company's total assets consists of current assets and long term assets. The total assets gives some indication of the size of the firm. Therefore it is frequently used as a normalizing factor (like in indicators 1,2,3,5 of Altman's indicators).

The current assets can or will typically be turned into money fairly fast. The firm's liabilities consists of current liabilities and long term debt. The current liabilities include short term loans (less than one year due), accounts payable, taxes due, etc. The working capital is current assets minus the current liabilities. It is an indication of the ability of the firm to pay its short term obligations. If it is too negative, the company might default on some payments. The firm's total assets is financed by a) the total liabilities and b) the shareholders' equity [therefore the name "balance sheet," since the total assets have to exactly equal the sum of the two items in (a) and b)]. The shareholders' equity consists of the capital raised in share offerings and the retained earnings. The retained earnings means the accumulation of the firm's earnings since the firm's inception. The shareholders' equity is also called the book value of the firm. Even though it is based on the historical costs (plus adjustments through depreciation/amortization) of the firm's assets and liabilities, rather than market values, it has been a very useful indicator in assessing the financial health of a firm. Retained earnings is a related and similarly useful indicator. The firm's earnings is also an important indicator. Highly negative earnings indicate that the firm is losing its competitiveness and that geopardizes its survival. Another related, widely used indicator is the cash flow. It is less prone than earnings to management manipulation. In addition, it measures directly the ability of the firm to generate cash to retire debt. The rationale behind Altman's fourth indicator is the following. The firm can issue and sell new shares in the market to repay its debt. A large market capitalization (relative to the total debt) indicates a high capacity to perform that.

Finally, the firm's sales is an indication of the health of its business. However, this indicator is probably the least effective among the five Altman indicators, because sales to total assets can vary a lot from industry to industry. Tam and Kiang (1992) considered the problem of bank failure prediction. They compared between several methods: MDA, LR, K-Nearest Neighbor (KNN), ID3 (a decision tree classification algorithm), single-layer network and multilayer network. For the case of one-year-ahead, the multilayer network was the best, while for the case of two-year-ahead, LR was the best. When they used a leave-one-out procedure instead of a hold-out sample, the multilayer network was the clear winner (for both forecast horizons). KNN and ID3 were almost always inferior to the other methods.

Salchenberger *et al.* (1992) considered the problem of predicting thrift failures. They compared NN with LR. The NN significantly outperformed the LR. For example for 18-months ahead prediction the LR achieves 83.3–85.4% accuracy (depending on some threshold), whereas the NN achieves 91.7%. Coats and Fant (1993) compared between NN and MDA. They obtained a classification accuracy in the range of 81.9 to 95.0% for the NN (depending on the horizon: from three-years ahead to less than a year-ahead) and in the range of 83.7 to 87.9% for the MDA (also depending on the horizon). Kerling and Poddig (1994) compared NN with MDA for a database of French firms for a three-year-ahead forecast. The NN achieved a prediction accuracy in the range of 85.3–87.7% compared to 85.7% forMDA. Kerling tested several cross-validation procedures and early-stopping procedures in a follow-through study . Altman *et al.* (1994) applied NN and MDA to a large database of 1000 Italian firms for one-year ahead prediction. The comparison yielded no decisive winner, though MDA was slightly better.

Boritz and Kennedy (1995) compared between a number of techniques, including different NN training procedures, LR and MDA, using the indicators chosen by Altman and those chosen by Ohlman. The results of the comparison are inconclusive. Fernandez and Olmeda (1995) compared NN with MDA, LR, MARS and C4.5 (two well known methods that are based on the CART decision tree algorithm) on Spanish banks (no horizon is specified). The NN obtained 82.4% accuracy compared with 61.8–79.4% for the competing techniques. Alici (1995) used principal component analysis and self-organizing maps for the input selection phase, together with a skeletonization step for the NN. He achieved an accuracy in the range of 69.5 to 73.7% (dependinding on some parameter variation), compared with 65.6% for MDA and 66.0% for LR for a database of UK firms (no horizon is mentioned). Leshno and Spector (1996) used a novel NN architectures containing cross-terms and cosine terms and achieved prediction accuracy for the two-years-ahead case in the range of 74.2-76.4% (depending on the order of the network), compared with 72% for the linear perceptron network. Lee *et al.* (1996) propose hybrid models. Specifically, they tested combinations of the models MDA, ID3, self-organizing maps and NN. They applied their study to the problem of default prediction of Korean firms.

Back *et al.* (1996) propose the use of genetic algorithms for input selection, to be used in conjunction with multilayer networks. They applied their method to data covering the periods one to three years before the creditworthiness, where it obtains significant improvement over MDA and LR. Kiviluoto use self-organizing maps on an extensive database of Finnish firms (horizon is not specified) and show that it obtains comparable results to MDA and learning vector quantization (in the range from 81 to 86%). Kaski developed a novel self-organizing map procedure based on the Fisher metric and applied it also to a number of Finnish firms.

Zhang *et al.* (1999) compared between NN and LR and employed a five-fold cross-validation procedure, on a sample of manufacturing firms (horizion is not specified). They used Altman's five financial ratios plus the ratio current assets/current liabilities as inputs to the NN. The NN significantly outperformed LR with accuracy of 88.2% versus 78.6%. Piramuthu *et al.* (1998) developed a technique to construct symbolic features, to be inputed to a multilayer network. They applied their technique to a collection of Belgian firms (no forecast horizon is mentioned), where they obtained an accuracy of 82.9% versus 76.1% for the nontransformed input case. They applied it also to a problem of one- and two-year ahead default prediction for US banks. They get superior results and significantly outperform the nontransformed input case. Piramuthu (1999) applies a similar input selection technique in conjunction with decision tree classifiers. Martinelli *et al.* (1999) compared between two decision tree algorithms, C4.5 and CN2 and NN on a database of Brazilian firms. C4.5 outperform the other methods. Yang *et al.* (1999) used probabilistic NNs (PNNs), which essentially implement the Bayes classification rule. They tested it on firms in the oil sector. The results were mixed: PNN tied with the multilayer networks, but with a particular preprocessing step MDA was the best.

McKee and Greenstein (2000) developed a method based on decision trees and applied it to a number of US firms for one year ahead forecast. Their method obtains better results thanNN and MDA for Type II error, but worse results for Type I error. Fan and Palaniswami propose the use of Support Vector Machines (SVMs, 2000) for predicting bankruptcies among Australian firms and compared it with NN, MDA and Learning Vector Quantization (LVQ). SVM obtained the best results (70.35-70.90% accuracy depending on the number of inputs used), followed by NN (66.11-68.33%), followed by LVQ (62.50-63.33%), followed by MDA (59.79-63.68%). These reviewed papers are just a sample of what has been done on the topic of NN default prediction.

THE EMPIRICAL MODEL

Artificial neural networks are information processing systems whose structure and function are motivated by the cognitive processes and organizational structure of neurobiological systems. The basic components of these networks are highly interconnected processing elements called neurons that work independently in parallel. Synaptic connections are used to carry messages from one neuron to another and the strength of these connections varies. These neurons store information and learn meaningful patterns by strengthening their interconnections. When a neuron receives a certain number of stimuli and when the sum of the received stimuli exceeds a certain threshold value, it fires and transmits the stimulus to adjacent neurons.

$$(x_1, x_2, x_3, \dots , x_n)$$

A single artificial neuron is the basic element of the neural network. It is comprised of several inputs and one output, y and can be written as:

$$Y = f(x_i, w_i) \qquad (1)$$

Where, w_i are the weights (that is, parameters) of the activation function, f, which maps any real input into a usually bounded range, often [0, 1] or [-1, 1]. This function may be linear or nonlinear such as the hyperbolic tangent, logistic, threshold, Gaussian and so on. If the activation function is linear, then (1) may be written as:

$$y = \sum x_i w_i = W^T X \quad \text{where} \quad 1 \le i \le m \qquad (2)$$

Where $w = [w_i]$ is the vector of weights assigned for each input, x_i, $x = [x_i]$ and ($i = 1, 2, ..., m$). Having several neurons with the same sets of inputs but with different outputs, the neuron layer can be constructed. Assuming that the activation function is linear, the output vector, y, can be derived from input data, x, as the weighted sum of the inputs:

$$y = W^T X$$
$$\text{where} \quad W^T = [w_{ij}] \tag{3}$$

Where the vector, $y = [y_j]$, consists of n outputs and is the transposed matrix [nxm] of weights. The layer of neurons shown by (3) is the simplest network.

Multilayer networks are formed by cascading a group of single layers. In a three-layer network, for example, there is an input layer, an output layer and a hidden layer. The nodes of different layers are densely interconnected through direct links. At the input layers, the nodes receive the values of input variables and multiply them through the network layer by layer. The middle layer nodes are often characterized as feature detectors. The number of hidden layers and the number of nodes in each hidden layer can be selected arbitrarily. The initial weights of the connections can be chosen randomly.

By applying the back propagation algorithm (which is the most popular method of supervised learning), the computed output is compared to the known output. If the generated output is correct, then nothing more is necessary. If the computed output is incorrect, then the weights are adjusted to make the computed output closer to the known output.

The training procedure runs in a certain number (usually determined by the ANN constructor) of iterations due to the algorithm below. Steps 1-6 are:

- Desire the ANN architecture.
- Provide the training set, that is, the statistical data for each input variable, x_i and desired values, z_j, for all output variables, y_j, which are to be standardized.
- Determine initial weights. They may be random from a certain interval or defined by the ANN constructor.
- Generate the output values, y_j, by ANN.
- Calculate errors between desired and generated outputs for each variable as:

$$\delta^{(k)} = Z - y^{(k)}, \; Z = [z_1]_{n\times 1} \tag{4}$$

  where: $\delta^{(k)}$ is the vector of errors, that is, differences between the outputs actually performed and the desired outputs; $y^{(k)}$ is the vector of the outputs actually performed (in the kth iteration); Z is the vector of desired outputs (j = 1, 2, ..., n) and k is the number of iterations in the training procedure.
- Modify the weight matrix for the next iteration according to the formula:

$$W^{(k+1)} = W^{(k)} + \eta^{(k)} \delta^{(k)} x^T \tag{5}$$

  Where: W^k is the matrix [nxm] of weights generated in the kth iteration and η^k is the acceleration coefficient (in the kth iteration).

The training procedure (beginning from Step 4 above) is continued:

- For a certain number of iterations;
- Until the defined convergence criterion is satisfied;

$$\delta_j < \gamma_j$$

- Until the network starts to simulate the system behavior properly, for example, where: γ_j is the maximal error determined by the ANN constructor for each output variable; and
- Until the ANN constructor decides to change the neural network architecture.

The aim of the training procedure is to minimize the cost function. The most commonly used cost function is a square of the error terms and can be written as:

$$Q = \sum_{j=1}^{n} (z_j - y_j)^2 \tag{6}$$

If the neural network solves a problem, it will have found a set of weights that produces the correct output for every input. The described algorithm can be used for ANN and consists of one layer since, in the multilayer networks, it is possible to evaluate errors (as in (4)) only for the output layer. The desired values for the hidden layer are not known, so the back propagation algorithm is used to propagate errors from output nodes backward until errors are calculated for all neurons. The errors in all layers except the output layer are estimated by applying the formula:

$$\delta_i^{l(k)} = \sum_{j=1}^{K^{l+1}} \delta_j^{(l+1)(k)} w_{ji}^{(l+1)(k)} \tag{7}$$

Where: $\delta_i^{l(k)}$ is the error regarding the ith neuron in the lth layer obtained in the kth iteration and K^l is the number of neurons in the lth layer. When $\delta_i^{l(k)}$ are known for all neurons, then all weights in ANN are modified according to (5). This process is continued for a large number of cases or time series until the net gives the correct output for a given input. The entire collection of cases learned is called a training sample. In most real-world problems, the neural network is never 100% correct. After the neural network learns up to the error threshold, then the weight adaptation mechanism is turned off and the net is tested on known cases that it has not seen before.

MODEL SIMULATION FOR CREDITWORTHINESS

Artificial neural networks are often used to solve pattern classification problems. Bankruptcy prediction is an example of such problems and neural networks are used to decide which category that the bankrupt or non-bankrupt firms fall into. Evaluating the firm's creditworthiness is similar to the bankruptcy prediction problem since the bank makes the decision after analyzing the firm's financial situation.

In Bangladesh, India and Pakistan, many commercial banks have been founded in the 1999s. These banks evaluate and determine the creditworthiness of companies by applying different methods of analysis. They also base their decisions on the kinds of credit involved (that is, consumer credit, investment credit, credit for current activity and so on) and the value of the credit. Generally, the decisions are made in two stages: after examining the legal framework and after examining the merits of the case.

The choices of variables are of two kinds: the choice on the firm population and the choice on descriptive variables of firms. The choices of the firm population mainly depend on sample size. Indeed, ANN needs a lot of examples to learn, but the number of bankrupt firms is not unlimited and the authors have to make a careful selection in order to keep sufficient number of data. Selection by branches of activity, geographic sector, firms size of the firm and observation period are limited by the size of the samples needed by ANN.

In this investigation, data was used from 75 small companies that applied for credit for current activity for the period September 1999 to June 2005. The value of the credit varied from $150 to $1,500 in monthly installments, to be repaid in less than five years. All enterprises analyzed by the bank was divided into two groups denoted as A and B. Group A consisted of 50 firms that obtained credit and Group B consisted of 25 companies that were denied credit during the procedure of creditworthiness determination. To consider a company as a potential borrower, the bank requires the following information:

- The characteristic of the activity provided by the company;
- The value of the credit and its appropriation;
- The proposed period of utilization for the credit and date of redemption;
- The expected effects of the undertaking;
- The participation of the internal and external sources in financing the undertaking;
- The form of legal hedge or security of the credit;
- The value of liabilities to other banks and institutions;
- Their pertinent information such as financial statements from the current period and from the previous year.

On the basis of this information, four input variables (neurons) were constructed and used in the creditworthiness analysis made by neural nets: x_1 is the monthly repayment of the loan and interest; x_2 is the average net monthly profit; x_3 is monthly liabilities to other banks and institutions; and x_4 is the evaluation of the firm's financial situation.

The variable x_4 represents a descriptive feature. For these experiments, it was transformed quantitatively, rating the economic and financial situation as: very good = 5; good = 4; satisfying = 3; poor = 2; and bad = 1. The output layer contained only one element representing the bank's decision. This neuron, in the training set, equaled 1 if the firm obtained credit and zero if the bank denied credit. Of course, values of output elements generated by neural nets belong to the interval [0, 1] and they rarely equal 1 or zero. Therefore, to classify the firm as A or B, it was necessary to define the threshold for generated outputs. Assume (Odnom and Sharda, 1990), That:

$$\tilde{y}_j = \begin{cases} 1 & \text{if } y_j \geq 0.5 \\ 0 & \text{if } y_j < 0.5 \end{cases} \tag{8}$$

Where: y_j is the value of generated output and $\tilde{y}_j$ is the value (zero or 1) representing the bank's decision.

The numerical experiments were conducted in two stages: training and testing. During training procedures, the desired values of outputs were provided. In the testing set, there was no such information. Since the testing procedure evaluates the accuracy of the classification made by the network, then the testing set should contain cases that were not presented in the training set. Thus, both stages require two separate sets of data.

Having a limited number of observations, decide next how to divide them into training and testing sets. Since there were 75 observations, assume that the training set contains 62 firms and the testing sample consists of 13 data. The training set should consist of cases that represent all possible situations, but it is hard to say how many A and B firms should be present in that set.

To study the effect of proportion between the number of firms obtaining credit and the number of firms denied credit, create three proportions for the training and testing set compositions. The first factor level was a 'A' proportion of positively (A) and negatively (B) evaluated firms. This means that the training set contained 31 firms denoted as A and 31 firms denoted as B (that is, 31/31) while the number of data in the testing set was 7/6. The second level was approximately B. In fact, there were 42/20 in training and 9/4 in testing sets. The third level was approximately C, that is, 50/12 and 10/3 for both data sets, respectively.

By analyzing the number of observations required for the training and testing set compositions, it is obvious that to investigate the influence of sample structures on the accuracy of classification, at least 76 A firms and 44 B firms are necessary. Thus, to construct training and testing samples with a certain number of observations, generate additional data for the A and B firms on the basis of the actual credit application forms. All real and generated observations were numerated. By applying sampling without a replacement scheme, select three pairs of the training and testing sets that contain a certain number of A and B companies. However, the testing samples were constructed in such way that they contained only data regarding the real firms that were analyzed by the bank.

In this investigation, consider three neural networks with one or two hidden layers, denoted as NN^1, NN^2 and NN^3. The activation function for the output layer was linear and, for the other layers, introduced logistic transfer functions.

Each ANN was trained on the basis of three training sets during 5,000, 10,000, 20,000, 40,000 and 100,000 iterations. After the whole set of experiments, 15 estimated neural models were tested. To evaluate the accuracy of these models, three types of errors were calculated in both stages. The percentage of general classification error is:

$$\xi = \frac{N_1 + N_2}{D + Z} * 100\% \qquad (9)$$

Where: N_1 is the number of B firms (denied by the bank) that were classified as A firms by the neural net; N_2 is the number of firms classified by the bank as A firms that were classified as B firms by the neural net; D is the number of all enterprises classified by the bank as A firms in the training or testing set; and Z is the number of all companies classified by the bank as B firms in the training or testing set. The percentage of error of first kind that describes the share of misclassified firms by the neural network B firms (classified by ANN as A) in the total number of enterprises evaluated by the bank as B firms is:

$$\xi_1 = \frac{N_1}{D} * 100\% \qquad (10)$$

The percentage error of second kind that describes the share of misclassified firms by the neural network A firms (classified by ANN as B) in the total number of enterprises evaluated by the bank as A firms is:

$$\xi_2 = \frac{N_2}{D} * 100\% \qquad (11)$$

The percentage share of correctly classified firms by neural network can be evaluated as:

$$P = 100 - \xi = \left(1 - \frac{N_1 + N_2}{D + Z}\right) * 100\% \qquad (12)$$

It is worth mentioning that from the bank's point of view, the error of first kind, ξ_1, is crucial since it is more costly to classify a B firm as an A firm (because it may cause this enterprise to not pay off its loan) than to misclassify an A firm.

EMPIRICAL RESULTS

The results of these experiments are presented in tables that contain ratios and the number of misclassified firms for all training Table 1 with result sets 1-3 and testing samples from Table 2-4. Analyze the efficiency of the neural networks, taking into account the proportion of training and testing sets, the network architecture and the number of iterations.

The general classification errors in the training procedure (Table 1) for the set A are less than 10% for all estimated neural networks. The number of correctly classified firms rises with the number of iterations. Usually, nets (trained on that sample) will classify A firms better. For the model NN^3, trained during 5,000 iterations, the highest error of second kind, ξ_2, is observed. This means that the nets have problems recognizing A firms when the number of training procedures is less than 10,000. Notice that the more complicated the network (such as NN^3), more iterations are required for proper classification.

ANN, trained Table 1 during 100,000 iterations on the basis of the B set, misclassified only one and every time, the same firm. After detailed analysis of the financial situation of that particular enterprise, the bank classifying that company as a B firm had to take into account other factors than the four defined input variables. Since considering values of input variables only, the bank should allow credit to this firm.

Table 1: Results of training the NN

Training set	Net architecture	No of iterations*	ξ_1	ξ_2	ξ	P
Result set 1						
A	NN^1	5	(4) 12.90	(1) 03.22	(5) 08.06	91.94
		10	(3) 09.68	(1) 00.00	(3) 04.84	95.16
		20	(2) 06.45	(1) 00.00	(2) 03.23	96.77
		40	(3) 09.68	(1) 00.00	(3) 04.84	95.16
		100	(1) 03.22	(1) 00.00	(1) 01.61	98.39
A	NN^2	5	(3) 09.68	(1) 03.22	(4) 06.46	93.54
		10	(3) 09.68	(1) 00.00	(3) 04.84	95.16
		20	(2) 06.45	(1) 00.00	(2) 03.23	96.77
		40	(3) 09.68	(1) 00.00	(3) 04.84	95.16
		100	(2) 03.22	(1) 00.00	(2) 03.23	96.77
A	NN^3	5	(1) 03.22	(5) 16.13	(6) 09.68	90.32
		10	(3) 09.68	(2) 06.45	(5) 08.08	91.92
		20	(3) 09.68	(1) 00.00	(3) 04.84	95.16
		40	(3) 09.68	(1) 00.00	(3) 04.84	95.16
		100	(2) 06.45	(1) 00.00	(2) 03.23	96.77
Result Set 2						
B	NN^1	5	(3) 15.00	(1) 00.00	(3) 04.84	95.16
		10	(2) 10.00	(1) 00.00	(2) 03.22	96.78
		20	(2) 10.00	(1) 00.00	(2) 03.22	96.78
		40	(2) 10.00	(1) 00.00	(2) 03.22	96.78
		100	(1) 05.00	(1) 00.00	(1) 01.61	98.39
B	NN^2	5	(3) 15.00	(1) 00.00	(3) 04.84	95.16
		10	(2) 10.00	(1) 00.00	(2) 03.22	96.78
		20	(2) 10.00	(1) 00.00	(2) 03.22	96.78
		40	(2) 10.00	(1) 00.00	(2) 03.22	96.78
		100	(1) 05.00	(1) 00.00	(1) 01.61	98.39
B	NN^3	5	(5) 25.00	(1) 00.00	(5) 08.06	91.94
		10	(2) 10.00	(1) 00.00	(2) 03.22	96.78
		20	(6) 30900	(1) 00.00	(6) 09.68	90.32
		40	(2) 10.00	(1) 00.00	(2) 03.22	96.78
		100	(1) 05.00	(1) 00.00	(1) 01.61	98.39
Result set 3						
C	NN^1	5	(6) 50.00	(1) 02.00	(7) 01.29	88.71
		10	(3) 25.00	(2) 04.00	(5) 08.06	91.94
		20	(3) 25.00	(2) 04.00	(5) 08.06	91.94
		40	(4) 33.33	(2) 04.00	(6) 09.68	90.32
		100	(3) 25.00	(1) 02.00	(4) 06.45	93.55
C	NN^2	5	(7) 58.33	(1) 02.00	(8) 12.90	87.10
		10	(6) 50.00	(1) 02.00	(7) 11.29	88.71
		20	(3) 25.00	(2) 04.00	(5) 08.06	93.55
		40	(3) 25.00	(1) 02.00	(4) 06.45	93.55
		100	(2) 16.67	(1) 02.00	(3) 04.84	95.16
C	NN^3	5	(4) 33.33	(1) 02.00	(5) 08.06	91.94
		10	(4) 33.33	(2) 04.00	(6) 09.68	90.32
		20	(3) 25900	(2) 04.00	(5) 08.06	91.94
		40	(3) 25.00	(2) 04.00	(5) 08.06	91.94
		100	(2) 16.67	(2) 04.00	(4) 06.45	93.55

Key: ‘denotes numbers are in thousands. Number of misclassified firms for all in parentheses

Table 2: Results of training ANN, trained on the basis of the sample A

Training set	Net architecture	No of iterations*	ξ_1	ξ_2	ξ	P
Result set 1						
A	NN^1	5	(1) 00.00	(3) 42.86	(3) 07.69	76.92
		10	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		20	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		40	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		100	(1) 00.00	(1) 14.29	(1) 07.69	92.31
A	NN^2	5	(1) 00.00	(3) 42.86	(3) 23.08	76.92
		10	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		20	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		40	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		100	(1) 16.67	(1) 14.29	(2) 07.69	84.62
A	NN^3	5	(1) 00.00	(4) 57.14	(4) 30.77	69.23
		10	(1) 00.00	(3) 42.86	(3) 23.08	76.92
		20	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		40	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		100	(1) 00.00	(1) 14.29	(1) 07.69	92.31

Continue Table 2

Training set	Net architecture	No of iterations*	ξ_1	ξ_2	ξ	P
Result Set 2						
B	NN^1	5	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		10	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		20	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		40	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		100	(1) 00.00	(2) 22.22	(2) 15.38	84.62
B	NN^2	5	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		10	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		20	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		40	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		100	(1) 16.67	(2) 22.22	(2) 15.38	84.62
B	NN^3	5	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		10	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		20	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		40	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		100	(1) 00.00	(2) 22.22	(2) 15.38	84.62
Result Set 3						
C	NN^1	5	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		10	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		20	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		40	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		100	(1) 00.00	(2) 20.00	(2) 15.38	84.62
C	NN^2	5	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		10	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		20	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		40	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		100	(1) 16.67	(2) 20.00	(2) 15.38	84.62
C	NN^3	5	(1) 00.00	(5) 50.00	(5) 38.46	61.54
		10	(1) 00.00	(5) 50.00	(5) 38.46	61.54
		20	(1) 00.00	(5) 50.00	(5) 38.46	61.54
		40	(1) 00.00	(5) 50.00	(5) 38.46	61.54
		100	(1) 00.00	(4) 40.00	(4) 30.77	69.23

Key: See key note in Table 1

Table 3: Results of training ANN, trained on the basis of the sample B

Training set	Net architecture	No of iterations*	ξ_1	ξ_2	ξ	P
Result set 1						
A	NN^1	5	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		10	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		20	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		40	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		100	(1) 00.00	(1) 14.29	(1) 07.69	92.31
A	NN^2	5	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		10	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		20	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		40	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		100	(1) 16.67	(1) 14.29	(1) 07.69	92.31
A	NN^3	5	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		10	(1) 00.00	(2) 28.57	(2) 15.38	84.62
		20	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		40	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		100	(1) 00.00	(1) 14.29	(1) 07.69	92.31
Result set 2						
B	NN^1	5	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		10	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		20	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		40	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		100	(1) 00.00	(1) 11.11	(1) 07.69	92.31
B	NN^2	5	(1) 00.00	(3) 33.33	(1) 23.08	76.92
		10	(1) 00.00	(3) 33.33	(1) 23.08	76.92
		20	(1) 00.00	(3) 33.33	(1) 23.08	76.92
		40	(1) 00.00	(3) 33.33	(1) 23.08	76.92
		100	(1) 16.67	(1) 11.11	(1) 07.69	92.31
B	NN^3	5	(1) 00.00	(1) 33.33	(1) 23.08	76.92
		10	(1) 00.00	(2) 33.33	(2) 23.08	76.92
		20	(1) 00.00	(1) 33.33	(1) 23.08	76.92
		40	(1) 00.00	(1) 33.33	(1) 23.08	76.92
		100	(1) 00.00	(1) 11.11	(1) 07.69	92.31

Continue Table 3

Result set 3

Training set	Net architecture	No of iterations*	ξ_1	ξ_2	ξ	P
C	NN^1	5	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		10	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		20	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		40	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		100	(1) 00.00	(2) 20.00	(2) 07.69	84.62
C	NN^2	5	(1) 00.00	(5) 50.00	(5) 38.46	61.54
		10	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		20	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		40	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		100	(1) 16.67	(2) 20.00	(2) 15.38	84.62
C	NN^3	5	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		10	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		20	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		40	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		100	(1) 00.00	(2) 20.00	(2) 15.38	84.62

Key: See key note in Table 1

Table 4: Results of training ANN, trained on the basis of the sample C

Training set	Net architecture	No of iterations*	ξ_1	ξ_2	ξ	P
Result set 1						
A	NN^1	5	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		10	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		20	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		40	(1) 16.67	(1) 14.29	(2) 15.38	84.62
		100	(3) 50.00	(1) 14.29	(4) 30.77	69.23
A	NN^2	5	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		10	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		20	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		40	(1) 00.00	(1) 14.29	(1) 07.69	92.31
		100	(3) 50.00	(1) 14.29	(4) 30.77	69.23
A	NN^3	5	(6)100.00	(4) 14.29	(7) 53.85	46.15
		10	(4) 66.67	(4) 14.29	(5) 38.46	61.54
		20	(4) 66.67	(4) 14.29	(5) 38.46	61.54
		40	(4) 66.67	(4) 14.29	(5) 38.46	61.54
		100	(3) 50.00	(2) 14.29	(4) 30.77	69.23
Result set 2						
B	NN^1	5	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		10	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		20	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		40	(1) 00.00	(3) 33.33	(3) 23.08	76.92
		100	(1) 25.00	(3) 33.33	(4) 30.77	69.23
B	NN^2	5	(1) 00.00	(1) 33.33	(1) 23.08	76.92
		10	(1) 00.00	(1) 33.33	(1) 23.08	76.92
		20	(1) 00.00	(1) 33.33	(1) 23.08	76.92
		40	(1) 00.00	(1) 33.33	(1) 23.08	76.92
		100	(2) 50.00	(1) 33.33	(4) 38.46	61.54
B	NN^3	5	(4)100.00	(3) 33.33	(7) 53.85	46.15
		10	(4)100.00	(3) 33.33	(7) 53.85	46.15
		20	(4)100.00	(3) 33.33	(7) 53.85	46.15
		40	(4)100.00	(3) 33.33	(7) 53.85	46.15
		100	(4)100.00	(3) 33.33	(7) 53.85	46.15
Result set 3						
C	NN^1	5	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		10	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		20	(3)100.00	(4) 40.00	(7) 53.85	46.15
		40	(3)100.00	(4) 40.00	(7) 53.85	46.15
		100	(3)100.00	(4) 40.00	(7) 53.85	46.15
C	NN^2	5	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		10	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		20	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		40	(1) 00.00	(4) 40.00	(4) 30.77	69.23
		100	(3)100.00	(4) 40.00	(4) 30.77	69.23
C	NN^3	5	(4)100.00	(4) 40.00	(7) 53.85	46.15
		10	(4)100.00	(4) 40.00	(7) 53.85	46.15
		20	(4)100.00	(4) 40.00	(7) 53.85	46.15
		40	(4)100.00	(4) 40.00	(7) 53.85	46.15
		100	(4)100.00	(4) 40.00	(7) 53.85	46.15

Key: See key note in Table 1

The classification errors obtained for the nets trained on the basis of the C data set Table 1 are higher than for the models estimated on the samples A and B, though after 100 iterations, general classification errors are not bigger than 6.45%. Errors, ξ_1, are relatively high because there were only 12 companies classified by the bank as B firms in the training set. However, neural nets made mistakes in classification more often than for other training sets.

The testing samples contain observations of input variables (x_1, x_2, x_3, ... x_n), only. During the testing procedure, the already trained neural networks generate values of the output elements, y_j, which are transformed into by applying (8). These outputs are compared to the actual decisions made by the bank.

By analyzing errors obtained for the models trained on the sample A Table 2, general classification errors are small for the testing sample A(less than 8% for nets NN^1 and NN^3, trained during 100,000 iterations and 15.38 percent for the model NN^2). For the sample B, the lowest errors equal 15.38% and, for the sample C, the majority of errors exceeded 30%. In all experiments but one, the error of first kind, ξ_1, is zero. This means that all (except one) B firms were correctly classified by ANN.

Testing the neural networks trained on the sample B Table 3, the perfect recognition of B firms and the relatively poor classification of A firms are observed. Also, errors are even smaller than for ANN, trained on the basis of the training set A.

Neural networks trained on the basis of the sample C Table 4 generates the highest errors for all testing sets and misclassification for both groups of firms. The neural network with two hidden layers seems to produce the highest errors. For instance, the general classification errors obtained for the net NN^3 belong to the interval [38.4, 53.85%]. This model misclassified all B firms in the testing sets B and C. For the testing set A, the errors of first kind belong to the interval [50, 100%]. Networks NN^1 and NN^2 produce better (testing) results for the models that were less accurate in the training procedure.

By analyzing the testing results from the point of view of the testing sample structure, the best results were obtained for the A set for all neural networks. For the testing sample B, the best results were obtained for the net trained on the B set, while testing sample C created the highest errors in all experiments.

CONCLUSION

This investigation shows that the structure of the samples influences the results of classification more than the architecture of ANN. The number of iterations required for proper classification depends on the number of hidden elements. The sets described as A and B are the most promising for training and testing since the smallest errors were obtained for the structure of the samples. It is hard to explain the reason for the relatively poor classification of B firms during the training process and for their perfect recognition during the testing procedure for samples A and B. It seems that this result is strictly connected with the empirical data since such results were not obtained in previous research Witkowska. It is worth mentioning that ANNs may become a bit over-zealous and, in an extreme case, the neural network might even learn the training sample completely thereby losing the capacity to transfer the acquired knowledge to other data records not included in the training set. Neural networks are the models that learn from the experience. Thus, the data sets used in the training process are extremely important. It is necessary to have a representation of all probable phenomena, requiring a huge number (at least 200-600) of observations to be used in the training process. In this research, only 75 actual firms were used for observation when applying for credit. Since this number is too small to investigate the influence of the sample structure on the results of classification, additional data was generated to perform these experiments.

To construct the ANN system, which determines the creditworthiness of enterprises, it is necessary to gather information concerning the firm's economic and financial condition as well as conduct profound analysis of their functioning environment. This investigation only used basic or available data since banks refused access to their documents to protect clients' confidentiality. Also, information was not obtained if the loans were repaid. Therefore, the ANN classification was made on a limited (in comparison with firm evaluation made by banks) data set. However, regardless of the small number of observations, the results of these experiments show that back propagation algorithm can be successfully used as an alternative method in a bank's decision-making procedure.

APPENDIX-A
POLICY GUIDELINES AND REMARKS

After evaluation of whole work, three main guidelines can be presented here:

- The hybridization of the training within the PMC framework: The RPG algorithm showing limits first of all was the subject of work tending to improve it by creation of new algorithms or simply by modification of existing algorithms (ex. CASCOR). In 2002, a new way was open with the use of external tools of RNA domain, the Genetic Algorithms, used in hybridization with the traditional algorithm: Dorsey, Edmister and Johnson (2003). If in the field of failure forecasting this approach is still recent, it was already the subject of much research in other fields and showed its capacities, to improve the traditional algorithm.

- The use of "new" networks: A second observation, relating to the development, is the enlargement of the type of network used. Indeed, the first experiments used mainly the PMC, but in 2001 and 2003, a new type of network already well-known appears in the domain of bankruptcy forecast. Two studies, Martin-del Brio and Serrano-Cinca (2001) and Kiviluoto and Bergius (2003) used self-organizing maps of KOHONEN, which seem to show satisfactory results. The architecture network field also seems to arrive at a phase of maturity, which makes it possible to test it in many implementations.
- The study of observation data: Work has until now used the same data that those discussed within the framework of traditional tools, but the appearance of particular characteristics of the ANN should see evolving data used. Thus a new way of studies is to observe the periods of data, the choice of the financial ratios and the introduction of new types of qualitative data.

Through this study, we can make some remarks:

- The greatest part of the experimentations make comparisons with traditional statistical forecast models such as Logistic Regression and Recursive Partitioning, but rare are the studies making a comparison between different neural networks.
- The study concludes with the superiority of Neural Networks. On research comparing of different studies Neural Networks with traditional statistics tools, 80% reach this conclusion.
- The observation data are heterogeneous from one experiment to another. The size constraint of the sample, for the use of Neural Networks prevents a voluntary step about the choice of the observation period, the scope of activity or the size of the firm. Comparisons between studies are thus difficult.
- The ratios used for the implementations are never chosen specifically for a neural network application. The variables are extracted from traditional studies on bankruptcy forecast, or from the existing literature. This method limits the potential contribution of neural networks to bankruptcy prediction.
- The specific abilities of Neural Networks are not retained. They are only used as substitute to traditional statistical tools, only taking advantage of the possibility, they don't ask any hypothesis about the distribution variables and the separability of the problem. But the specificities such as determining a relationship between a large amount of data, or dealing with incomplete data, are not highlighted.

After many years of research, a first step has been reached, proving the capabilities of neural networks in bankruptcy prediction. Indeed, the greatest part of the authors agree to present their work as preliminary studies, that prove the interest of using neural networks for this type of classification. So this research show the superiority of neural networks on Logit Analysis ; some other researchers also concluded this proposition, Tan (2001) concludes to the superiority of neural networks on Probit analysis, Coats and Fant (1999) present them as superior to MDA. The major part of the studies conclude these new techniques give equal or sometimes even better results than traditional tools.

APPENDIX-B
COMPARISON OF ANN MODEL WITH LOGISTIC REGRESSION

It is easy to observe that logistic regression model is a special case of neural network model. If we choose for each of the activation g_γ, $g_{,1}$ … , g_{m-1} the identity activation:

$$g(z) = z$$

We have:

$$E\{Y_i\} = \beta_0 + \beta_1 H_{i1} + \beta_{m-1} H_{i,m-1} \quad \text{(a)}$$

and:

$$H_{ij} = \alpha_{j0} + \alpha_{j1} X_{i1} + \ldots + \alpha_{j,p-1} X_{i,p-1} \quad \text{(b)}$$

Substitution of (a) into (b) and rearranging yields:

$$E\{Y_i\} = \left[\beta_0 + \sum_{j=1}^{m-1} \beta_j a_{ji}\right] + \left[\sum_{j=1}^{m-1} \beta_j a_{ji}\right] X_{i1} + \ldots + \left[\sum_{j=1}^{m-1} \beta_j a_{j,p-1}\right] X_{i,p-1} \quad \text{(C)}$$

$$= \beta_0^* + \beta_1^* X_{i1} + \ldots + \beta^*_{p-1} X_{i,p-1} \quad \text{(f)}$$

Where:

$$\beta_0^* = \beta_0 \sum_{j=1}^{m-1} \beta_j a_{j0}$$

In General:

$$ß_K^* = \sum_{j=1}^{m-1} ß_j a_{jk} \quad \text{for } k=1,...,p-1 \qquad (h)$$

The neural network with identity activation functions thus reduces to the standard logistic regression model. Hence, Artificial Neural network Model for forecasting is nothing just Generalization of Logistic Regression

REFERENCES

Alici, Y., 1995. Neural networks in corporate failure prediction: The UK experience, In Proc. Third Int. Conf. Neural Networks in the Capital Markets, A.N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend, (Eds.), London, UK, pp: 393-406.

Altman, E., 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, J. Finance, pp: 589-609

Altman, E., G. Marco and F. Varetto, 1994. Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks, J. Banking and Finance, pp: 505-529.

Back, B., T. Laitinen and K. Sere, 1996. Neural networks and genetic algorithms for bankruptcy predictions, Expert Sys. Applied, pp: 407-413.

Beaver, R., 1966. Financial ratios as predictors of failure, Empirical Research in Accounting: Selected Studies J. Accounting Res., pp: 71-111.

Boritz, J. and D. Kennedy, 1995. Effectiveness of neural network types for prediction of business failure, Expert Sys. Applied, pp: 504-512.

Coats, P. and L. Fant, 1993. Recognizing financial distress patterns using a neural network tool, Financial Manage., pp: 142-155.

Dimitras, A., S. Zanakis and C. Zopounidis, 1996. A survey of business failures with an emphasis on prediction methods and industrial applications, Eur. J. Oper. Res., pp: 487-513.

Fan A. and M. Palaniswami, 2000. A new approach to corporate loan default prediction from financial statements, In: Proc. Computational Finance/ Forecasting Financial Markets Conf. CF/FFM-2000, London (CD), UK.

Fernandez, E. and I. Olmeda, 1995. Bankruptcy prediction with artificial neural networks, Lect. Notes Comput. Sci., pp: 1142-1146.

Kerling, M. and T. Poddig, 1994. Klassifikation von Unternehmen mittels KNN, In Neuronale Netze in der Ökonomie, H. Rehkugler and H. G. Zimmermann, Eds. München, Germany.

Kiviluoto, K., 1998. Predicting bankruptcies with the self-organizing map, Neurocomputing, pp: 191-201.

Lee, K., I. Han and Y. Kwon, 1996. Hybrid neural network models for bankruptcy predictions, Decision Support Sys., pp: 63-72.

Leshno, M. and Y. Spector, 1996. Neural network prediction analysis: The bankruptcy case, Neurocomputing, pp: 125-147.

Martinelli, E., A. de Carvalho, S. Rezende and A. Matias, 1999. Rules extractions from banks' bankrupt data using connectionist and symbolic learning algorithms, In: Proc. Computational Finance Conf., New York.

McKee, T.E. and M. Greenstein, 2000. Predicting bankruptcy using recursive partitioning and a realistically proportioned data set, J. Forecasting, pp: 219-230.

Moody's Quantitative Risk's Public Firm Risk Model. [Online]. Available: www.moodysqra.com

Odom, M. and R. Sharda, 1990. A neural network model for bankruptcy prediction, In Proc. Int. Joint Conf. Neural Networks, San Diego, CA.

Ohlson, J., 1980. Financial ratios and the probabilistic prediction of bankruptcy, J. Accounting Res., pp: 109-131.

Piramuthu, S., 1999. Feature selection for financial credit-risk evaluation decisions, Informs J. Computing, pp: 258-266.

Piramuthu, S., H. Raghavan and M. Shaw, 1998. Using feature construction to improve the performance of neural networks, Manage. Sci., pp: 416-430.

Salchenberger, L., E. Cinar and N. Lash, 1992. Neural networks: A new tool for predicting thrift failures, Decision Sci., pp: 899-916.

Tam, K. and M. Kiang, 1992. Managerial applications of the neural networks: The case of bank failure predictions, Manage. Sci., pp: 416-430.

Vellido, A., P. Lisboa and J. Vaughan, 1999. Neural networks in business: A survey of applications (1992-1998), Expert Syst. Applied, pp: 51-70.

Wong, B., T. Bodnovich and Y. Selvi, 1997. Neural network applications in business: A review and analysis of the literature (1988-95), Decision Support Sys., pp: 301-320.

Yang, Z., M. Platt and H. Platt, 1999. Probabilistic neural networks in bankruptcy prediction, J. Business Res., pp: 67-74.

Zhang, G., M. Hu and B. Patuwo *et al.*, 1999. Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis, Eur. J. Oper. Res., pp: 16-32.