The Evaluation of Dutch Non-Life Insurance Companies

A Comparison of an Ordered Logit and a Neural Network Model

Bert Kramer[*]

SOM theme A: Structure, Control and Organization of Primary Processes

ABSTRACT

This paper describes the development of two models which determine the financial solidity of Dutch non-life insurers: an ordered logit model and a neural network model. Since bankruptcies of Dutch insurance companies are very rare, the annual assessment texts made by the supervisor are used to classify the companies into one of the three possible groups: strong, moderate, or weak. Both models use the same six variables, which were selected by means of a stepwise logistic selection procedure. These variables cover three aspects: solvency, profitability, and investments. Both models are estimated on a 1992 data set which contains 195 companies, and they are tested on a 1993 data set containing 193 companies. The ordered logit model correctly classified 85% of the test set, 95% of the strong companies and 85% of the weak companies are classified correctly. The neural network model correctly classified 86% of the test set, 96% of both the strong and the weak companies are classified correctly. A combination of the outcomes of both models leads to an overall score of 87%, with 97% of the strong and 96% of the weak companies classified correctly. Neither the ordered logit model nor the neural network model are able to adequately recognize moderate companies. Some possible reasons for the problems with moderate companies are given.

[*] Faculty of Management and Organization, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands, Phone +31503637178, Fax +31503633850, e-Mail e.l.kramer@bdk.rug.nl. I would like to thank the Verzekeringskamer for their support and for giving me access to the data.

1 Introduction

The insurance industry is subject to government regulation to safeguard the interests of the policyholders. In order to protect the interests of the policyholders, the Dutch government has established the insurance supervisory board (ISB, in Dutch: Verzekeringskamer). The primary goal of the ISB is to protect the policyholders against failures of insurance companies to meet their obligations. To reach this goal, the ISB assesses the solvency of the companies.

The evaluation of insurance companies is primarily based on annual reports which all Dutch insurance companies are obliged to hand in. These reports include a number of prescribed financial statements. The evaluation is done in three rounds. The verification round consists of a consistency check of the financial statements, and of a solvency check. With the solvency check, the solvency margin is compared with the statutory required margin. In the assessment round the financial statements are thoroughly analyzed by a (senior) employee. The final evaluation is performed by the account manager. The findings from the different rounds are written down in the assessment text. If the evaluation of a company leads to any doubt about the solidity of that company, then the account manager can decide to request more information or to visit the company for a more thorough examination.

An on-site inspection can give a better idea of the solidity of a company than a sole analysis of the financial statements. On-site inspections enable the ISB to make sure that the documents submitted reflect the actual situation, on the one hand, and to obtain additional prospective information about the business practices of the insurance company, on the other (Angerer, 1993, p.42). However, given the large number of companies which have to be evaluated (in 1992, 97 life and 391 non-life[2] insurers) and the limited number of employees (20), there is not enough time left for an annual on-site inspection of each company. Normally, if no problems are visible, a non-life company is visited once in five to ten years.

A computer based system that can identify insurers with financial difficulties can be a useful tool for scheduling the timing, intensity, and extent of a particular insurer's examination. A system which evaluates the financial solidity of insurance

[2] Of which 243 non-life insurance companies are seated in the Netherlands.

companies can be used to classify these insurers according to their degree of risk exposure. Given this exposure an examination schedule can be established. Such a system could take over the solvency check in the verification round and part of the assessment round. Moreover, such a system might be able to locate problem companies more quickly enabling the ISB to react faster.

In this paper, we describe the application of two different models to determine the financial solidity of Dutch non-life insurers: an ordered logit model and a neural network model. These models can be used to classify insurers according to their degree of risk exposure. The distinction between bankrupt and non-bankrupt companies will not be made here since bankruptcies of Dutch non-life insurance companies are very rare. The classification criterium and the data set will be described in the next section. A general description of ordered logit models will be given in section 3, followed by a description of neural networks in section 4. The resulting models will be described in section 5 and 6. The conclusions are given in section 7.

2 Classification Methodology and Data

Earlier applications of classification techniques for financial institutions usually distinguished bankrupt from non-bankrupt companies[3]. Most of these studies were based on USA-data. In the USA, there have been a number of bankruptcies which, although undesirable for supervisors and policyholders, is reasonable in a statistical sense. The most popular techniques in this field are multiple discriminant analysis and logit analysis[4].

Since bankruptcies of Dutch non-life insurance companies are very rare, and because of a lack of other objective classification criteria, we will have to resort to a somewhat subjective classification criterium. In this project, the classifications are based on the assessment texts for the companies. The major weak and strong points of a company are described in that text as value statements, criticisms, or indications of negative or positive trends. The companies are classified into one of the three

[3] See (BarNiv & McDonald, 1992) for a recent survey.

[4] See (Altman et al., 1981) for an excellent discussion of these techniques.

possible groups: strong (1), moderate (2), or weak (3). The classification depends on the number and the severity of the weak and strong points as given in the assessment text. Thus, we are trying to model the assessments made by the ISB.

A disadvantage of this approach is that the assessment text might not fully describe the actual situation of a company. The employees of the ISB might not write down their full opinion of a company, or some important aspects might not be known to the ISB. Also, for some companies the assessment text lacks clear value statements. For those companies the classification can be difficult and can become somewhat arbitrary. These problems are inevitable because of a lack of workable (more) objective criteria.

In 1992, 243 Dutch non-life insurance companies reported to the ISB. Of these 243 companies, 40 companies started their operations after 1990. Since 2-year growth-rates are used, these companies had to be excluded from the data set. Furthermore, all (=7) run-off companies had to be excluded because their specific character (such as, a premium income which is very close or equal to zero) leads to extremely large values for some financial ratios. Finally, one company had to be excluded from the data set because for a number of potentially important ratios the denominator was equal to zero. So the 1992 data set, which has been used to estimate the models, contains a total of 195 companies.

Before using a model in practice, it is preferable to know whether the performance of the model is sufficiently stable over the years. To test this, a 1993 data set has been used which contains all Dutch non-life insurance companies, excluding run-off companies and companies less than 3 years old, for which at least the second assessment round had been completed by February 20, 1995. The 1993 data set contains 193 companies.

Seventy variables were selected on the basis of their use in the assessment process at the ISB, and on their successful usage in previous studies (a.o., Trieschmann & Pinches, 1973; Harrington & Nelson, 1986; BarNiv & Raveh, 1989; BarNiv & McDonald, 1992; Brockett et al., 1994). These include financial ratios and their 1- and 2-year growth-rates, and some dummy variables. The main aspects covered by the variables are: solvency, profitability, investments, reinsurance, types of risk insured, technical provisions, premium growth and market penetration, and dependence on group-companies.

3 Ordered Logit Models

In this study, a polytomous extension of the standard (binary) logit model is used. With this model it is assumed that there is a natural ranking in the possible values of the dependent variable. This is clearly the case for this application: a strong financial solidity (1) is better than a moderate financial solidity (2), which is still better than a weak financial solidity (3).

Suppose that y_n can take on k **ordinally ranked values**, with $y_n = 1$ the "lowest" and $y_n = k$ the "highest". Furthermore, suppose that

$$\begin{aligned} P_{1n} &= F(X_n^{/}\beta) \\ P_{1n} + P_{2n} &= F(\alpha_2 + X_n^{/}\beta) \\ &\vdots \\ P_{1n} + P_{2n} + \cdots + P_{k-1,n} &= F(\alpha_{k-1} + X_n^{/}\beta) \\ P_{kn} &= 1 - F(\alpha_{k-1} + X_n^{/}\beta) \end{aligned}$$

where P_{in} is the conditional probability that choice i occurs for the nth observation, X_n is a vector of m independent variables, β is a m-length parameter-vector,

$$\alpha_{k-1} > \alpha_{k-2} > \cdots > \alpha_2 > \alpha_1 = 0$$

are threshold parameters, and

$$F(\alpha_i + X_n^{/}\beta) = \frac{1}{1 + \exp[-\alpha_i - X_n^{/}\beta]}$$

is the cumulative logistic function. This model is called the **ordered logit model**.

The motivation behind this model (McKelvey & Zavoina, 1975; Green, 1990, p.703-704) is that there is a continuous but unobserved variable Y_n which is a linear function of the X's and a stochastic standard logistic variable ϵ_n (μ=0, σ=1.81). Thus,

$$Y_n = -X_n^{/}\beta + \epsilon_n$$

In contrast to discriminant analysis, where normality is assumed, with the ordered logit model no assumptions are made about the distribution of the independent variables. The observed choice y_n is determined by the value of Y_n: $y_n = 1$ if $Y_n \leq 0$; $y_n = 2$ if $0 < Y_n \leq \alpha_2$; ...; $y_n = k - 1$ if $\alpha_{k-2} < Y_n \leq \alpha_{k-1}$; $y_n = k$ if $\alpha_{k-1} < Y_n$. The width of each range is fixed across observations. However, the probability of Y_n falling in each range will vary by observations as $X_n'\beta$ changes. The scaling variables $\{\alpha_2, ..., \alpha_{k-1}\}$ are estimated by the data rather than externally imposed. That is, both the β-vector and the α-vector are estimated by maximum likelihood.

4 Neural Networks

Neural networks are endowed with some special features which make them an interesting candidate to determine the financial solidity of insurance companies. A neural network does not start with an a priori model of the relationships between input and output. It is actually discovering the relationship between the input and output itself. This might be an advantage compared to the traditional statistical models like logit. The most commonly used type of neural networks, one-hidden-layer back-propagation neural networks, which are also used in this study, can be seen as a nonparametric optimization technique (Geman et al., 1992, p.40-41). The number of applications of neural networks has shown a rapid increase in recent years. A promising application of neural networks to early warning of insurer insolvency can be found in (Brockett et al., 1994).

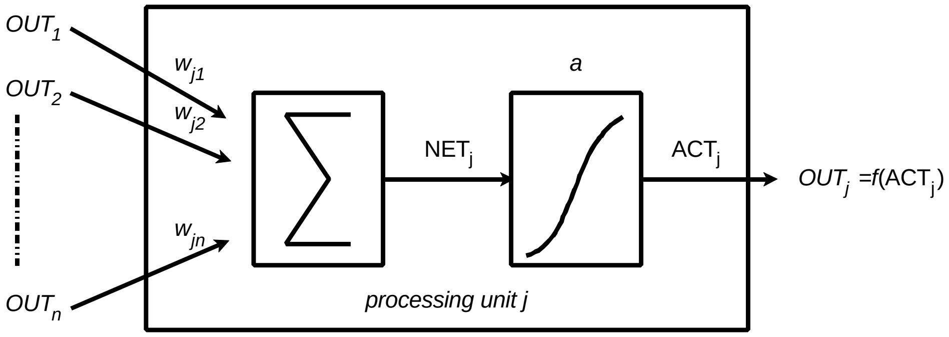
Seven important aspects of a neural network are (Rumelhart et al., 1992, p.57-63):

1. A *set of processing units* or neurons.
2. A *state of activation*.
3. An *output function* for each unit that maps its state of activation into an output.
4. A *pattern of connectivity* among units, that is, the weights for each of the connections.
5. A *propagation rule* for propagating patterns of activities through the network of connectives.
6. An *activation function* for combining the inputs impinging on a unit with the current state of that unit in order to produce a new level of activation for the unit.
7. A *learning rule* whereby patterns of connectivity are modified by

experience.

ad.1 *A set of processing units* or neurons. All of the processing of a neural network is carried out by these units. There is no executive or other overseer. There are only relatively simple units, each doing its own relatively simple job. A unit's job is simply to receive input from other units and, as a function of the inputs it receives, to compute an output value which it sends to other units. The system is inherently parallel in that many units can carry out their computations at the same time. Figure 1 shows a model of a processing unit.

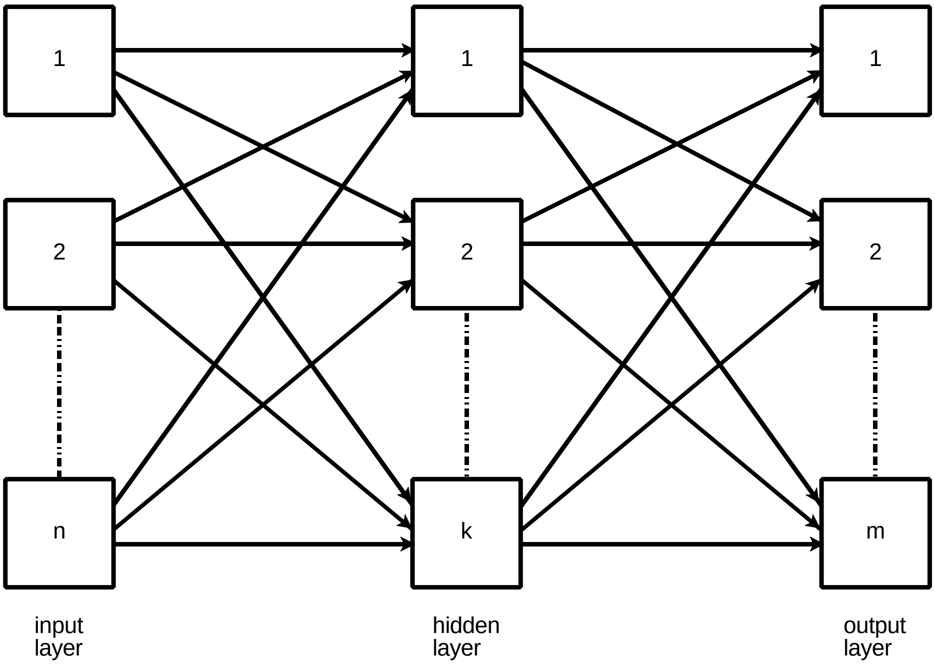
Figure 1: a processing unit



It is useful to characterize three types of units: input, output, and hidden. *Input units* receive inputs from sources external to the system under study. The *output units* send signals out of the system. The *hidden units* are those whose only inputs and outputs are within the system. They are not visible outside the system.

The units in a neural network are usually arranged in layers. Multi-layer networks may be formed by simply cascading a group of single layers: the output to one layer provides the input to the subsequent layer. A *layer* consists of a set of weights and the subsequent units that sum the signals they carry (Wasserman, 1989, p.22). Figure 2 illustrates the basic structure of a one-hidden-layer neural network with *n* input units, *k* hidden units, and *m* output units.

Figure 2: a one-hidden-layer neural network



The number of processing units in a layer will depend on the problem which has to be solved. The choice of the number of input- and output units for a specific problem is quite straight-forward. The choice of the number of units in the hidden layer(s) is, however, more difficult. There are only rules of thumb to help a researcher with this choice.

For most applications, only one hidden layer is needed. If there is no good reason to have more than one hidden layer, then you should stick to one. The training time of a network increases rapidly with the number of layers (Caudill, 1991, p.59).

ad.2 *The state of activation*. The level of activation of the units taken collectively represents the state of the system. It is convenient to look on the processing carried out by the system as the evolution of the system state. Activation of any particular unit induces or hinders the activation of units to which it is connected according to whether the interconnection is excitatory or inhibitory. The notion of activation per se may be viewed in two different ways (Khanna, 1990, p.13). First, the activation value of a unit indicates its degree of confidence that its associated feature is present or absent, as opposed to merely providing a yes/no answer regarding the presence or absence of a feature. Alternatively, the activation value of a unit might suggest the quantity of a feature that is present. The activation value is passed through a function to produce an output value.

ad.3 *Output function* ($OUT_j = f(ACT_j)$). Units interact by transmitting signals to other units. The strength of their signals, and therefore the degree to which they affect other units, is determined by their degree of activation. The output value can be seen as passing through a set of unidirectional connections to other units in the system. Associated with each unit, there is an output function *f* which maps the current state of activation to an output signal. Sometimes, the output level is exactly equal to the activation level of the unit. In other cases, the output function is some sort of threshold function so that a unit has no effect on another unit unless its activation exceeds a certain value. Alternatively, the output function might be a stochastic function in which the output of the unit depends in a probabilistic fashion on its activation values.

ad.4 *The pattern of connectivity* (w_{ji}). Units are connected to one another. It is this pattern of connectivity that constitutes what the system knows and that determines how it will respond to any arbitrary input. The total pattern of connectivity can be specified by defining the weights for each of the connections in the system. The *weight* or strength w_{ji} of a connection determines the amount of effect that unit *i* has on unit *j*.

Depending on the pattern of connectivity, two types of networks can be distinguished: nonrecurrent or *feedforward networks* and *recurrent networks*. *Feedforward networks* have no feedback connections, that is, they have no connections through weights extending from the outputs of a layer to the inputs of the same or previous layers. Feedforward networks have no memory; their output is solely determined by the current inputs and the values of the weights. *Recurrent networks* do contain feedback connections. In some configurations, recurrent networks recirculate previous outputs back to inputs; hence, their output is determined both by their current input and their previous outputs (Wasserman, 1989, p.19-20).

ad.5 *The rule of propagation* ($NET_j = \sum w_{ji} \ OUT_i$). All inputs to a unit are multiplied by their associated weights and summed to get the net input to that unit.

The net input to a unit, along with its current activation value determine its new activation value.

ad.6 *Activation function* (ACT_j = *a* (NET_j)). The activation function *a* combines the inputs impinging on a particular unit with the current state of the unit to produce a new state of activation. Whenever the activation value is assumed to take on continuous values, it is common to assume that *a* is a kind of sigmoid (i.e., S-shaped) function. In that case, an individual unit can saturate and reach a minimum or maximum activation value. With a logistic activation function, the limits of the output neuron are 0 and 1. With a hyperbolic tangent function, the limits are -1 and 1. If the problem involves learning about 'average' behavior, logistic activation functions work best, but if the problem involves learning about 'deviations' from the average, hyperbolic tangent works best (Klimasauskas, 1993, p.65).

ad.7 *The learning or training rule.* A network is trained so that application of a set of inputs produces the desired (or at least consistent) set of outputs. Each such input (or output) set is referred to as a vector. Training is accomplished by sequentially applying input vectors, while adjusting network weights according to a predetermined procedure. Training algorithms can be categorized as *supervised* and *unsupervised* training.

Supervised training requires the pairing of each input vector with a target vector representing the desired output; together these are called a training pair. Usually a network is trained over a number of such training pairs. An input vector is applied, the output of the network is calculated and compared to the corresponding target vector, and the difference (error) is fed back through the network and weights are changed according to an algorithm that tends to minimize the error. The vectors of the training set are applied sequentially, and errors are calculated and weights adjusted for each vector, until the error for the entire training set is at an acceptably low level.

In *unsupervised training* or clustering, the training set consists solely of input vectors. The training algorithm modifies network weights to produce output vectors that are consistent; that is, both the application of one of the training vectors or the application of a vector that is sufficiently similar to it will produce the same pattern of outputs. The training process, therefore, extracts the statistical properties of the training set and groups similar vectors into classes.

Back-propagation, the most successful of the current neural network algorithms, provides a systematic means for (supervised) training of multi-layer feedforward networks. A back-propagation network starts out with a random set of weights. The network adjusts its weights each time it sees an input-output pair. Each pair requires two stages: a forward pass and a backward pass. The *forward pass* involves presenting a sample input to the network and letting the activation of the units flow until they reach the output layer. The logistic function is normally used as activation function. There are, however, many functions that might be used; the back-propagation algorithm requires only that the function be everywhere differentiable. The logistic function satisfies this requirement. In the following, a logistic activation function is assumed. For simplicity, it is also assumed that the output level is equal to the activation level of the unit.

During the *backward pass*, the network's actual output vector (from the forward pass) is compared with the target output vector and error estimates are computed for the output units. The weights of the connections between the (last) hidden layer and the output layer *o* can be adjusted in order to reduce those errors. This adjustment is accomplished using the error signal (i.e., the target minus the actual output) multiplied by the derivative of the activation function, which is equal to OUT_j (1 - OUT_j) for a logistic activation function:

$$\delta_{jo} = \mathrm{OUT}_{jo} (1 - \mathrm{OUT}_{jo}) (\mathrm{TARGET}_{jo} - \mathrm{OUT}_{jo})$$

Thereafter, the delta value for unit *j* in the output layer *o*, δ_{jo}, is multiplied by the output value from the source unit *i* in (hidden) layer *q* for the weight in question, OUT_{iq}. This product is in turn multiplied by the learning rate coefficient η and the result is added to the weight from unit *i* in layer *q* to output unit *j*:

$$\Delta w_{jio} = \eta \, \delta_{jo} \, \mathrm{OUT}_{iq}$$

The learning-rate coefficient η determines the average size of the weight changes (the step size).

Since hidden layers have no target vector, the training process described above cannot be used for adjusting the weights between subsequent hidden layers (if more than one hidden layer exists), and for adjusting the weights between the input layer and the (first) hidden layer. Instead, back-propagation trains the hidden layers by propagating the output error back through the network layer by layer, adjusting weights at each layer. These weights now operate in reverse, passing the delta value from the output layer back to the hidden layer(s). Each of these weights is multiplied by the delta value of the unit to which it connects in the subsequent layer. The delta value needed for unit *j* in layer *p* is produced by summing all such products for the subsequent layer *q* and multiplying by the derivative of the activation function:

$$\delta_{jp} = OUT_{jp} (1 - OUT_{jp}) (\Sigma_{i} \delta_{iq} w_{ijq})$$

For each unit in a given hidden layer, the deltas must be calculated, and all weights associated with that layer must be adjusted. This is repeated, moving back towards the input layer by layer, until all weights are adjusted.

5 Outcomes of the Ordered Logit Model

To determine the variables to be included in the model, a stepwise selection procedure was used. The decision to include a variable in the model is based on the log-likelihood and on the likelihood-ratio test[5]. The selection of the explanatory variables was purely based on statistical grounds, and no a priori assumptions were made on which variables were to be included, while also the sign of the coefficients for the different variables was not considered in the selection procedure. The final model is given in table 1. The variables are defined in the appendix.

[5] The Wald test, which uses the t-ratio, is an alternative to the likelihood ratio test. However, Hauck and Donner (1977) examined the performance of the Wald test and found that it behaved in an aberrant manner, often failing to reject when the coefficient was significant. They recommended that the likelihood ratio test be used. Jennings (1986) also analyzed the adequacy of inferences in logistic regression based on Wald statistics. His conclusions are similar to those of Hauck and Donner. Therefore, the likelihood ratio test was used in this study.

Table 1: the ordered logit model

Variable	Coefficient	Standard Error
Constant	7.69	1.58
SOLVRA	- 4.21	0.69
PATNEP	- 7.50	2.18
G2SURRSM	0.085	0.150
PROPERTY	12.02	14.46
FISNTPSM	1.97	0.82
G1COMBRA	- 2.32	1.20
α_2	2.50	0.66

α_2 is the threshold parameter from the ordered logit model (with k = 3). A negative (positive) coefficient indicates a positive (negative) relation between that variable and the financial solidity of a company, since a higher score coincides with a lower financial solidity (1=strong, 2=moderate, 3=weak).

As could be expected, the first two variables which were included in the model are a solvency ratio and a profitability ratio. Solvency and profitability are widely recognized to be important aspects of the financial solidity of an insurance company (Oosenbrug, 1994). The included variables cover three aspects of financial solidity: solvency (SOLVRA and G2SURRSM), profitability (PATNEP and G1COMBRA), and investment-portfolio (PROPERTY and FISNTPSM).[6]

For some variables the coefficient shows a counter-intuitive sign. However, the coefficients are influenced by the other variables and their coefficients. If we look

[6] Oosenbrug (1994) gives the following, non-exhaustive, list of aspects which influence the financial solidity of insurance companies: 1. solvency, 2. profitability, 3. investment-portfolio and -strategy, 4. quality of reinsurance and reinsurance-strategy, 5. quality and types (lines) of risks insured, 6. quality of the technical provisions, 7. premium growth and market penetration, 8. quality and multiformity of management, and 9. types of distribution-channels and their quality. Other aspects which are sometimes mentioned are: organization type (stock or mutual) and dependence on group-companies.

at the correlations between the included variables and the actual class on a univariate basis, then FISNTPSM and G1COMBRA have the opposite sign. The only variable that shows a counter-intuitive sign both in the model and on a univariate basis is G2SURRSM. An improvement in the solvency surplus ratio is a positive development. However, G2SURRSM is especially large if the solvency surplus was around zero two years ago. So, if G2SURRSM is large this usually indicates that the company was close to insolvency two years ago, and that it is recovering from this weak solvency position. However, a company which was almost insolvent two years ago often cannot be called strong yet. Furthermore, companies with a deteriorating solvency position will usually have low values for SOLVRA and PATNEP. For 'normal' values of G2SURRSM, this variable will be dominated by SOLVRA and PATNEP. G2SURRSM only dominates SOLVRA and PATNEP if its value is large.

The results of the model for the 1993 data set are given in table 2. The outcome *j* as predicted by the model for the *i*th company is that outcome *j* for which Prob(y_i = *j*) has the largest value. In table 2, SCORE is the percentage of companies that has been classified correctly. For instance, SCORE for strong companies equals 139/147=94.6% and TOTAL SCORE equals (139+3+22)/193=85.0%.

Table 2: Actual and predicted outcomes for the ordered logit model, 1993 data

	Predict				
Actual	1	2	3	TOTAL	SCORE
1	139	5	3	147	94.6%
2	11	3	6	20	15.0%
3	1	3	22	26	84.6%
TOTAL	151	11	31	193	85.0%

For both the strong companies and the weak companies the model performs quite well. The number of moderate companies that has been classified correctly is, however, low. A reason for the high number of misclassifications of moderate companies might be that the group of moderate companies is very heterogeneous. A number of companies are classified as moderate although they are sufficiently solvent because they have certain peculiarities which either make them more risky or which makes it difficult to assess their risk profile. A more detailed model might be necessary to adequately classify moderate companies.

Furthermore, the maximum likelihood estimation method equally weighs all types of errors. Only the difference between the actual and the predicted outcome counts. Since the class of strong companies is, by far, the largest, the maximum likelihood method has a tendency to better classify strong companies than weak and, especially, moderate companies. The group of moderate companies is the smallest group. This tendency in favor of strong companies and, in particular, against moderate companies can be reduced by putting more weight on misclassifying moderate companies. This can, for instance, be done by increasing the relative share of moderate companies in the data set. This would lead to a lower total score, and to a higher score for moderate companies.

Neither a more detailed model, nor adjusting the weights will be tried here. The performance of the model on the most important class, the weak companies, is quite good. Almost all weak companies get a high priority, and most companies which get a high priority (i.e., the companies which the model classifies as weak) really are weak. Furthermore, only one weak company gets a low priority. These are useful properties for a system which will be used as a tool to support the supervision by the ISB. In this way, the ISB can concentrate on the most important cases. When we try to improve the performance on the class of moderate companies, then this might lead to a deterioration in the performance on weak and strong companies. This might cause a rise in the number of strong companies and a drop in the number of weak companies with a high priority. This is undesirable.

Finally, maybe in reality there are only two classes, strong and weak, and the moderate class is actually a "don't know" class. That is, either the employees of the ISB are not sure whether a company is strong or weak, or the assessment text is just not clear enough. Therefore, a company is classified as moderate while it is actually strong or weak. However, the model might be able to recognize such a company as strong or weak. In this case it is not strange that the model performs bad on moderate companies.

6 Outcomes of the Neural Network Model

6.1 Network design

In total, 4 network designs have been tested. For all designs, the input of the network consists of the 6 significant financial ratios of the ordered logit model. Neural network theory does not provide us with an efficient procedure to determine the 'best' subset of variables for a certain application, other than running the network on each possible subset of variables. The stepwise logistic regression methodology can be considered a shortcut for this very time-consuming procedure. According to Brockett et al. (1994, p.412), the stepwise logistic regression procedure is a computationally efficient approximation to the results that might have been obtained with the 'all subsets' methodology for neural networks. The output consists of three scores between 0 to 1, one for each possible class.

The activation function for the hidden and output neurons is a hyperbolic tangent function, for the input neurons a linear function is used. For both functions the limits are -1 and 1. The input data are normalized such that they fall within the range of the activation function. The 1st and 2nd design use the absolute minimum and maximum of the data for normalization. Sometimes using the absolute minimum and maximum causes problems, especially if the data contains outliers (Lawrence, 1994, p.207-209). If the network uses the absolute minima and maxima that include the outliers, it will be difficult for it to tell the difference between facts with values close together. The network will understand extreme cases better, but it will not perform as well in the typical scenario. By adjusting the minimum and maximum values this bias can be reduced. In the 3rd and 4th design the minima and maxima are adjusted such that about 95% of the data will be within the new range.

In the 2nd and 3rd design the number of hidden neurons is set equal to 6. In the two other designs, we begin with a small number of hidden neurons and add more during the training process if the network is not learning. This is called the constructive approach to optimizing the hidden layer size (Klimasauskas, 1993, p.69). In the 1st design we start with 4 hidden neurons and add a neuron if the RMS error does not decrease over 100 training runs. In the 4th design we start with 5 hidden neurons and add a neuron if the RMS error does not decrease over 500 training runs.

The data are unevenly distributed, there are many more strong companies than moderate and weak companies. With a high learning rate, the network will tend to learn that all companies are strong. A technique which works well for many unevenly distributed data sets is to set the learning rate to a very low value (CSS, 1993, p.8-27). The network will then train very slowly, because it's taking very small steps to the answer, but it will have the advantage of not moving too quickly in the direction of the majority solution. In order to be able to find the patterns for the minority data, low learning rates are used in this study. In the 1st network design, the learning rate is set to 0.25, and in the other designs it is set to 0.2. Table 3 gives an overview of the network designs.

Table 3: Overview of the network designs

design[1]	learn rate	# hidden neurons	add hidden[2]	min-max[3]
1	0.25	4	100	absolute
2	0.2	6	-	absolute
3	0.2	6	-	adjusted
4	0.2	5	500	adjusted

[1] In this table, only the design parameters which differ between the different designs are given. The common design parameters can be found in the main text.
[2] Add a neuron to the hidden layer if the RMS error does not decrease over the specified number of training runs.
[3] This column indicates whether the normalization of the input variables is based on their absolute minimum and maximum or on adjusted values.

To avoid overfitting, the available data is split into three parts (Weigend et al., 1990, p.196-197). The 1992 data set is divided into two sets: a training set, used for determining the values of the weights, and a testing set, used for deciding when to stop training. The performance on the testing set is monitored. As long as this performance improves, training continues. When it ceases to improve, training is stopped. Eighty percent (156 companies) of the 1992 data set is used for training, the remaining twenty percent (39 companies) is used for testing. Since the performance of the network might be influenced by the composition of the training and the testing set, 10 random 80-20 partitions have been generated. So, for each design 10 networks will be trained, which results in a total of 40 networks.

To estimate the expected performance of the network in the future, the 1993 data set is used as a prediction set. This data set is strictly set apart and is neither used in training, nor is it used to determine the termination of the training process.

The optimal network for this application is the network with the best performance on the prediction set.

6.2 Training, testing, and prediction results

For each design (1,...,4) and random partition (a,...,j), the network has been trained over 10000 runs. The training process is stopped before run 10000 if the network has perfectly learned either the training or the testing set. After the training had stopped, the development over the runs of the RMS error on the training set and the number of errors on the testing set have been analyzed. If a network still showed improvement over the last runs, then training was continued for another 5000 runs. If a network performed very badly on the testing set, and if this performance did not show any improvement over the runs, then the weights of the network were randomized again and the network was retrained with these new starting values.

For each of the 40 design/partition set-ups, the networks from the runs with the minimum number of errors on the testing set were selected. The final choice between the resulting 40 networks will be based on the performance of these networks on the prediction set. The percentages of companies correctly classified by the different networks for the prediction set are given in table 4.

Table 4: Percentages of correctly classified companies, 1993 data

partition										
design	a	b	c	d	e	f	g	h	i	j
1	79	78	86	80	85	79	83	78	86	76
2	79	77	86	76	76	77	82	83	74	77
3	34	32	33	32	36	48	27	54	43	24
4	33	33	86	34	27	32	34	44	33	32

Design 3 and 4 (apart from partition c) show very bad performance on the prediction set. With these designs the majority of companies is classified as weak and very few companies are classified as strong. Since the majority of the companies (147 out of 193) is actually strong, the scores are very low. Where this bias towards moderate and weak companies comes from is not clear. A number of networks from the other designs have a bias towards strong companies. However, this does not lead to very low scores.

There are 4 networks with the maximum score of 86%: 1c, 1i, 2c, and 4c. Of these networks, network 1i is preferred. This network has the highest score for weak companies (96%, compared to 77% for 1c and 4c and 81% for 2c), and it also has the lowest number of companies misclassified by two classes (i.e., weak companies misclassified as strong plus strong companies misclassified as weak, for 1i this is 5, for 1c and 2c it is 7, and for 4c it is 9).

Network 1i has, by far, the largest number of hidden neurons (43). Given the number of training facts, the number of hidden neurons is very high. There are more weights than training facts. However, the risk of memorization of the training set, causing poor results during testing, has been removed by using the three data sets. The fact that a network with such a large hidden layer gives the best generalization indicates that the underlying problem structure might be very complex. The more complex the underlying problem structure is, the more hidden neurons are needed to adequately capture this structure. However, the 2nd and 4th best generalizing networks (2c and 4c) only have 6 hidden neurons, and the 3rd best generalizing network (1c) has 9 hidden neurons. So maybe the (undetected) 'perfect' network does not need that many hidden neurons. The classification table for network 1i is given in table 5.

Table 5: Actual and predicted outcomes for network 1i, 1993 data

	Predict				
Actual	1	2	3	TOTAL	SCORE
1	141	2	4	147	95.9%
2	13	0	7	20	0%
3	1	0	25	26	96.2%
TOTAL	155	2	36	193	86.0%

6.3 A comparison of the neural network and the ordered logit model

The scores for strong and weak companies are very high for network 1i. The percentages are higher than those for the ordered logit model (table 2). Especially the score for weak companies is very convincing. The overall score of the neural network model is slightly higher than that of the ordered logit model (86% versus 85%).

The neural network completely fails to recognize moderate companies. Apart from the possible explanations given for the ordered logit model in section 5, an additional possible explanation for the neural network is, that, given the highest overall score, the final network was chosen on the basis of its performance on weak companies, and on the (lowest) number of misclassifications by two classes. Network 1i performed best on both criteria, while the other three networks with an overall score of 86% performed better on moderate companies.

To test to what extent the ordered logit model and the neural network model agree on their assessments, the average scores have been calculated for all companies. For these average scores, the regular and the Spearman correlation coefficients between the models have been calculated. The regular correlation coefficient equals 0.94, and the Spearman correlation coefficient, measuring the correlation of the ranks, equals 0.87. These values, which are very significant (<0.001, 2-tailed), indicate that there is a strong agreement between the models.

Although there is a strong agreement between the models, they disagree on the classification of a number of companies. Some companies are misclassified by one of the models, but not by the other. A combination of both models might, therefore, give an even better performance than the performance of the separate models. The misclassification of a company by the ordered logit model might be compensated by the correct classification of the neural network model and a misclassification by the neural network model might be compensated by the correct classification of the ordered logit model. To test this, the neural network scores for the different classes are first normalized, such that they sum to one for each company. This normalization is done by dividing each score by the sum of the three output scores for a company. Thereafter, these normalized scores are added to the probabilities for each company belonging to the different classes as given by the ordered logit model. The predicted class for a company is the class with the highest value of the sum of the probability according to the ordered logit model and the normalized score from the neural network model. The three scores for the three possible classes now range from 0 to 2, and they sum to 2. Thus, the outcomes of both models have an equal weight. The resulting classification table is given in table 6.

Table 6: Actual and predicted outcomes, sum of the ordered logit probability and the normalized neural network score, 1993 data

	Predict				
Actual	1	2	3	TOTAL	SCORE
1	142	0	5	147	96.6%
2	12	1	7	20	5.0%
3	1	0	25	26	96.2%
TOTAL	155	1	37	193	87.0%

The total score for the combination of models is higher than the total score for each model taken separately. For both the weak and the strong companies over 96% is correctly classified. These are very high percentages for an out of sample test, giving us some confidence in the usefulness of this combination of models. Just as with the separate models, the score for moderate companies is low.

6.4 Interpretation of the weights

Although it is difficult to explain why a particular conclusion is reached by a neural network, something can be said about the relative importance of the different input variables for the different output neurons. This can be done by calculating the strength of the relationship between each input and each output variable, which is measured by the following statistic (Yoon et al., 1994, p.502):

$$RS_{ji} = \frac{\sum_{k=0}^{n} (W_{ki} \cdot W_{jk})}{\sum_{i=0}^{m} |\sum_{k=0}^{n} (W_{ki} \cdot W_{jk})|}$$

where RS_{ji} is the relative strength between the *i*th input and the *j*th output variable, W_{ki} is the weight between the *k*th hidden neuron and the *i*th input neuron, and W_{jk} is the weight between the *j*th output neuron and the *k*th hidden neuron. In multivariate analysis, this approach is used frequently to determine the proportion of variation of one variable in relation to all the others.

This statistic measures the strength of the relationship between the *i*th input and the *j*th output variable to the total strength of all of the input and output variables. The result is the percentage of all output weights attributable to the given independent variable, excluding bias weights. Because we used three different output neurons for the three different classes, the relative importance of the different input neurons for the different output classes can be determined. The relative strengths for network 1i are given in table 7.

Table 7: Relative strengths between the input and the output variables, network 1i

	STRONG	MODERATE	WEAK
SOLVRA	0.5356	-0.0561	-0.4934
PATNEP	0.1814	-0.3944	-0.0733
G2SURRSM	0.0352	0.2455	-0.1120
PROPERTY	-0.0711	0.1257	0.0716
FISNTPSM	-0.1717	-0.1109	0.1753
G1COMBRA	-0.0049	-0.0673	-0.0743

The solvency ratio (SOLVRA) is clearly the most important factor for both the strong and the weak class. As could be expected, SOLVRA and strong are positively related, and SOLVRA and weak are negatively related. The profit ratio (PATNEP) is also positively related to strong and negatively related to weak: higher profits will lead to a higher score for strong, and a lower score for weak, other things equal. For the investment ratios (PROPERTY and FISNTPSM) the opposite is valid: they are negatively related to strong and positively related to weak. For these four variables, the directions of the relations are equal for the neural network and for the ordered logit model (see table 1). For the solvency-surplus growth-ratio (G2SURRSM) the direction of the relation for the neural network is opposite to the direction for the ordered logit model. A higher value will increase the score for strong, and decrease the score for weak in the neural network. The growth of the combined ratio (G1COMBRA) does not give a significant contribution to strong, but this ratio is negatively related to moderate and weak.

7 Conclusions

Six variables give a significant contribution to the determination of the financial solidity of a Dutch non-life insurance company. These variables cover the aspects solvency, profitability, and investments. Aspects which were not found to be significant in the model are, amongst others, dependence on group-companies, types of risk insured, and organization type (mutual or stock).

The ordered logit model correctly classifies 95% of the strong and 85% of the weak companies, while the neural network correctly classifies 96% of both the strong and the weak companies. These percentages are comparable to those reported in comparable studies, but where the distinction between the companies is based on whether or not they went insolvent. For both models, the number of correctly classified moderate companies is very low. Some possible explanations for this are given.

The overall performance of the neural network is slightly higher than that of the ordered logit model. This difference is mainly due to the better performance of the neural network on weak companies. A combination of both models shows an even better overall performance. The combined model correctly classifies 97% of the strong and 96% of the weak companies.

In spite of the counter-intuitive sign of some of the coefficients and the lack of theoretical criteria for the selection of the variables, the (combination of) models might be a useful tool to support the supervision of Dutch non-life insurance companies. First of all, the models perform quite well. Furthermore, the models would not take over the full evaluation process. They would only be used to establish priorities for the assessment round, which would then be performed by employees of the ISB. In this way, the verification round can be done quicker, and potential problem companies will be treated first in the assessment round. A thorough analysis by human experts will, however, still be necessary.

References

Altman, E.I., R.B. Avery, R.A. Eisenbeis and J.F. Sinkey (1981), *Application of Classification Techniques in Business, Banking and Finance* , Greenwich, CC, JAI Press.

Angerer, A. (1993), 'Insurance Supervision in OECD Member Countries,' in *Policy Issues in Insurance*, Paris, France, OECD, pp. 13-72.

BarNiv, R. and J.B. McDonald (1992), 'Identifying Financial Distress in the Insurance Industry: A Synthesis of Methodological and Empirical Issues,' *Journal of Risk and Insurance* , 59, pp. 543-574.

BarNiv, R. and A. Raveh (1989), 'Identifying Financial Distress: A New Nonparametric Approach,' *Journal of Business Finance & Accounting* , 16, pp. 361-383.

Brockett, P.L., W.W. Cooper, L.L. Golden and U. Pitaktong (1994), 'A Neural Network Method for Obtaining an Early Warning of Insurer Insolvency,', *Journal of Risk and Insurance* , 61, pp. 402-424.

Caudill, M. (1991), 'Neural Network Training Tips and Techniques,' *AI Expert*, January, pp. 56-61.

CSS (1993), *Brainmaker Professional User's Guide and Reference Manual* , 4th edition, Nevada City, CA, California Scientific Software.

Geman, S., E. Bienenstock and R. Doursat (1992), 'Neural Networks and the Bias/Variance Dilemma,' *Neural Computation*, 4, pp. 1-58.

Green, W.H. (1990), *Econometric Analysis*, New York, NY, Macmillan.

Harrington, S.E. and J.M. Nelson (1986), 'A Regression-based Methodology for Solvency Surveillance in the Property-Liability Insurance Industry,' *Journal of Risk and Insurance*, 53, pp. 583-605.

Hauck, W.W. and A. Donner (1977), 'Wald's Test as Applied to Hypotheses in Logit Analysis,' *Journal of the American Statistical Association* , 72, pp. 851-853.

Jennings, D.E. (1986), 'Judging Inference Adequacy in Logistic Regression,' *Journal of the American Statistical Association* , 81, pp. 471-476.

Khanna, T. (1990), *Foundations of Neural Networks*, Reading, MA, Addison-Wesley.

Klimasauskas, C.C. (1993), 'Applying Neural Networks,' in: R.R. Trippi and E. Turban (eds.), *Neural Networks in Finance and Investing* , Chicago, ILL., Probus, pp. 47-72.

Lawrence, J. (1994), *Introduction to Neural Networks: Design, Theory, and Applications*, 6th edition, Nevada City, CA, California Scientific Press.

McKelvey, R.D. and W. Zavoina (1975), 'A Statistical Model for the Analysis of Ordinal Level Dependent Variables,' *Journal of Mathematical Sociology*, 4, pp. 103-120.

Oosenbrug, A. (1994), 'Rating-Systemen en de Kwalificatie van Verzekeringsmaatschappijen,' in: L.A.A. van den Berghe, A. Oosenbrug, R. Kaas and H. Wolthuis (eds.), *Heterogeniteit in Verzekering*, Rotterdam, the Netherlands, Erasmus Insurance Center, pp. 259-277.

Rumelhart, D.E., G.E. Hinton and J.L. McClelland (1992), 'A General Framework for Parallel Distributed Processing', in P. Mehra and B.W. Wah (eds.), *Artificial Neural Networks: Concepts and Theory*, Los Alamitos, CA, IEEE Computer Society Press, pp. 56-82.

Trieschmann, J.S. and G.E. Pinches (1973), 'A Multivariate Model for Predicting Financially Distressed P-L Insurers,' *Journal of Risk and Insurance*, 40, pp. 327-338.

Wasserman, P.D. (1989), *Neural Computing: Theory and Practice*, New York, NY, Van Nostrand Reinhold.

Weigend, A.S., B.A. Huberman and D.E. Rumelhart (1990), 'Predicting the Future: A Connectionist Approach,' *International Journal of Neural Systems*, 1, pp. 193-209.

Weiss, S.M. and C.A. Kulikowski (1991), *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*, San Mateo, CA, Morgan Kaufmann.

Yoon, Y., T. Guimaraes and G. Swales (1994), 'Integrating Artificial Neural Networks with Rule-Based Expert Systems,' *Decision Support Systems*, 11, pp. 497-507.

Appendix: Variable description

FISNTPSM	fixed interest-bearing securities/(net technical provisions + minimum required solvency margin)

G1COMBRA	one-year growth of the combined ratio: (gross incurred claims/gross earned premiums)+((operating expenses + gross commissions)/gross booked premiums)
G2SURRSM	two-year growth of ((solvency margin - minimum required)/minimum required)
PATNEP	profit after taxes/net earned premiums
PROPERTY	property investments/invested capital
SOLVRA	solvency margin/minimum required solvency margin