Tilburg University

Application of Neural Networks to House Pricing and Bond Rating

Daniëls, H.A.M.; Kamp, B.; Verkooijen, W.J.H.

Publication date:
1997

Link to publication in Tilburg University Research Portal

Citation for published version (APA):
Daniëls, H. A. M., Kamp, B., & Verkooijen, W. J. H. (1997). *Application of Neural Networks to House Pricing and Bond Rating*. (CentER Discussion Paper; Vol. 1997-96). Operations research.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 12. May. 2021

APPLICATION OF NEURAL NETWORKS TO HOUSE PRICING AND BOND RATING

Hennie Daniels[1,2], Bart Kamp[1], and William Verkooijen[3]

[1] *Tilburg University, P.O. Box 90153, 5000 LE Tilburg,*
The Netherlands, Department of Economics,
Section Management Information Systems

[2] *Erasmus University Rotterdam,P.O. Box 1738*
3000 DR Rotterdam, The Netherlands
Rotterdam School of Management

[3] *ABN-AMRO Bank, 1186 AA Amstelveen, The Netherlands*
ISIS/ECE/Information Warehouse Facilities

Abstract

Feed forward neural networks receive a growing attention as a data modelling tool in economic classification problems. It is well-known that controlling the design of a neural network can be cumbersome. Inaccuracies may lead to a manifold of problems in the application such as higher errors due to local optima, overfitting and ill-conditioning of the network, especially when the number of observations is small. In this paper we provide a method to overcome these difficulties by regulating the flexibility of the network and by rendering measures for validating the final network. In particular a method is proposed to equilibrate the number of hidden neurons and the value of the weight decay parameter based on 5 and 10-fold cross-validation. In the validation process the performance of the neural network is compared with a linear model with the same input variables. The degree of monotonicity with respect to each explanatory variable is calculated by numerical differentiation. The outcomes of this analysis is compared to what is expected from economic theory. Furthermore we propose a scheme for the application of monotonic neural networks to problems where monotonicity with respect to the explanatory variables is known a priori. The methods are illustrated in two case studies: predicting the price of housing in Boston metropolitan area and the classification of bond ratings.

Keywords: Classification, Error estimation, Monotonicity, Finance, Neural-network models
JEL C52, C63

1. Introduction

There is a growing attention for neural networks as a tool for data analysis. To a certain extent, the popularity of neural networks compared to other statistical methods may be caused by the failure of statisticians to communicate their methodologies and algorithms to non-statisticians. The vast amount of accumulated statistical knowledge puts up a barrier for consumers of their methods. Neural networks, on the other hand, are in an embryonic phase, which means that the accumulated knowledge is relatively small. The language used within the neural network community is another factor which may explain the success of neural networks. However, the core problems of data analysis do not change when the techniques they are approached with are changed. Therefore, difficulties statisticians have run into will also affect neural network practitioners. The specification of a neural network involves not only a selection of the inputs, but also the selection of the various components of a network, such as the type of network to use, the squashing function, which error criterion, which learning algorithm, the number of hidden layers, and how many hidden units per layer. Once these network components have been specified, the neural network is confronted with the data.

In many classification and prediction problems in economics, data sets are small and special techniques are needed to reliably estimate the prediction error as well as to avoid overfitting. Especially when neural networks are applied to time series prediction, where some of the input series are non-stationary, overfitting is very likely (Verkooijen, 1996). But also in simpler classification tasks when no precautions are taken, overfitting may occur (Geman *et al.*, 1981). Another problem, much ignored, is the landing in local optima of the error function during the training process (Ripley, 1993). These practical issues are important factors that determine the success of neural network applications; therefore they require careful investigation. The impact of particular choices of the network components is largest in small sample problems, where statistical theory is not of much help.

The aim of this paper is to lay down the choices concerning the different aspects of neural network modelling mentioned above, and to establish a general network construction procedure. In particular we discuss how several techniques like cross-validation, weight decay and monotonicity analysis can be effectively combined to optimize the neural network. There are two main approaches to control the complexity and the flexibility of the neural network: model selection and regularization.

Model selection for neural networks involves choosing the number of hidden units, the connections, and the inputs. By regularization the neural network solution is smoothed by stop training or weight decay. The simplest approach is to stop training after a predetermined number of "epochs", which are complete presentations of the whole training set. It is obvious that this approach can only be suboptimal. A more realistic approach is to use a test set of data (set B) to indicate the error on ’unseen’ cases; these data may not be used during training. When the error on the test set starts to increase, training is terminated. To measure the degree of generalization, a third independent set (C, the validation set) is necessary to estimate the out-of-sample performance of the network. The sets B and C are not used in training the network, which incurs a loss of costly information for problems with limited data. In practice C should be taken as large as possible to ensure a low variance prediction error estimator, but on the other hand as much observations should be used to reliably estimate the network weights.

In many economic problems we expect a monotonic relation (but not necessarily linear) of the output variable with respect to some of the inputs. In both case studies presented in this paper, we computed the degree of monotonicity of the network with respect to all input variables. These monotonicity properties can be used to add natural bias to the neural network. This is another way of controlling the flexibility of the neural network, and reduces the danger of overfitting.

Another method to circumvent overfitting is weight decay learning. Weight decay enforces a reduction of the flexibility of a neural network by adding a penalty to the error term, which then becomes:

$$W(\lambda) = E + \lambda . \sum_i w_i^2 \qquad \textbf{(1)}$$

In this way weights are penalized for growing too large without reducing E much. The value of λ regulates the smoothness of the neural network. Large values of λ correspond to neural networks that are forced to generate smoother approximations at the cost of lower flexibility. If on the other hand the number of neurons in the hidden layer (N_h) is increased, the flexibility of the network is enlarged. The optimal network is constructed by deliberating the values of the parameters N_h and λ. In practice one strives to compute "optimal" combinations of λ and N_h such that we obtain the best possible fit in-sample and out-of-sample simultaneously (cf. table 3).

The remainder of this paper is organized as follows. In section 2 we describe a neural network model to predict the price index of houses as a function of the object characteristics: a so-called hedonic model. The process of model building based on 10-fold cross-validation is described in section 2.3. The performance of the final neural network model is compared with two linear models. The first model (M_1) is linear in both the parameters and variables, and the second model (M_2) is linear in the parameters, but also includes non-linearly transformed explanatory variables like in (Harrison *et al.*, 1978). Furthermore we obtain a measure for the degree of monotonicity of each of the explanatory variables in the final neural network. In section 3 we describe the construction of a neural network model for classification of bond ratings. We applied 5-fold cross validation to estimate the prediction error. The performance of the neural network model is compared with a linear model. In section 3.5 the degree of monotonicity is calculated.

2. Modelling the Hedonic House Price in Boston

2.1 Description of the case study

In this section we want to compare the performance of a linear model, a modified linear model and a neural network model to estimate the house price in Boston. The techniques described in the introduction are applied in the neural network construction procedure.

The basic principle of the hedonic approach to economics is that each consumer good is regarded as a bundle of characteristics for which an implicit valuation exists (Janssen, 1992). (Harrison *et al.*, 1978) regard each house as a bundle of characteristics and the price of each house as reflective of the value of its characteristics. So the price of a house is estimated by the equation:

$$H^p = g(x_1,...,x_q) \qquad \textbf{(2)}$$

where each x_i denotes a characteristic of the house. In this case there is no theoretical knowledge of what the function *g* should look like. We therefore employ a data driven approach to specify the equation (2). In the original study of (Harrison *et al.*, 1978) their interest was to estimate the impact of air pollution on the prices of houses. However, the model may serve many different purposes. For example in many countries the local tax authorities require house values to calculate the amount of property tax due. The data are of cross-sectional type, i.e. the attributes are measured across various suburbs of Boston at a particular time point.

The explanatory variables in equation (2) are listed in table 1.

Table 1. Definition of model variables.

symbol	definition
CRIM	per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 sq.ft
INDUS	proportion of non-retail business acres per town
CHAS	Charles River dummy variable (=1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centers
RAD	index of accessibility to radial highways
PTRATIO	pupil-teacher ratio by town
LSTAT	% lower status of the population
H^p	Median value of owner-occupied homes in $1000’s

The data set consists of 506 instances and was taken from the StatLib library which is maintained at Carnegie Mellon University. The basic data, which are also listed in (Belsley *et al.*, 1980), are a sample of census tracts in the Boston Standard Metropolitan Statistical Area in 1970. The matrix of cross-correlations between all attributes is presented in table 8 in the appendix. Useful information can be extracted from it at a glance. The correlation matrix suggests, for example, that the number of rooms per dwelling (RM) and the % lower status of the population (LSTAT) are important determinants of the housing value. The direction of influence corresponds with common sense: more rooms will in general result in a higher housing value and a high percentage of lower status of the population will decrease the value of a house. the relationship between NOX and INDUS, which says that industrial areas are more polluted than rural areas, is another example.

2.2 Linear models

For the Boston house price problem, there is no theoretical knowledge that prescribes a specific functional form of the relationship between H^p and the other attributes. An obvious start to specify the model, is to try to fit a linear model (in both parameters and variables) to the data:

$$H^p = \alpha_0 + \alpha_1 \mathrm{CRIM} + \alpha_2 \mathrm{ZN} + \alpha_3 \mathrm{INDUS} + \alpha_4 \mathrm{CHAS} + \alpha_5 \mathrm{NOX} + \alpha_6 \mathrm{RM} + \alpha_7 \mathrm{AGE} + \alpha_8 \mathrm{DIS} + \alpha_9 \mathrm{RAD} + \alpha_{10} \mathrm{TAX} + \alpha_{11} \mathrm{PTRATIO} + \alpha_{12} \mathrm{B} + \alpha_{13} \mathrm{LSTAT} + \epsilon \qquad (\mathbf{M_1})$$

The OLS estimates of the parameters are shown in table 10 in the appendix. The attributes AGE and INDUS have no significant (at a 5% level) effect on H^p, so they are left out. Although the signs of the estimated coefficients correspond to what is expected from economic or common sense knowledge, graphical inspection of plots of the residuals against each attribute and against estimated H^p provides evidence of a misspecified functional form. The usual response to this phenomenon is to transform the variables which seem to affect the dependent variable in a non-linear way by some parametric function (e.g., log x, , or x^2), suggested by the various plots. In this way, it can be quite time consuming to find the right functional form; the investigator has to search manually after a suitable functional form, using the data at hand. In (Harrison *et al.*, 1978) the following linear model (in the parameters) is proposed and examined for fit.

$$\log(H^p) = \alpha_0 + \alpha_1 \mathrm{CRIM} + \alpha_2 \mathrm{ZN} + \alpha_3 \mathrm{INDUS} + \alpha_4 \mathrm{CHAS} + \alpha_5 \mathrm{NOX}^2 + \alpha_6 \mathrm{RM}^2 + \alpha_7 \mathrm{AGE} + \alpha_8 \log(\mathrm{DIS}) + \alpha_9 \log(\mathrm{RAD}) + \alpha_{10} \mathrm{TAX} + \alpha_{11} \mathrm{PTRATIO} + \alpha_{12} \mathrm{B} + \alpha_{13} \log(\mathrm{LSTAT}) + \epsilon \qquad (\mathbf{M_2})$$

The OLS estimates (with standard errors) of modelling MEDV by M_1 are presented in table 10 in the appendix. In contrast with model M_1 several log and square-transformations of the variables have been made. The effect of these transformations is an increase in R^2 to 0.81 (measured in back-transformed values), so a better fit is obtained. The estimated coefficients, the standard errors, and the corresponding t-values of model M_2 are presented in table 11 in the appendix.

A neural network offers an alternative to this manual transformation approach. It should be able to make an approximation to the data automatically, which is as good as M_2 or better. The next section investigates whether this is the case.

2.3 A neural network model

The foregoing results indicate that non-linearities are present in the house price equation. In this section a neural network is constructed to capture the non-linear behavior. The neural network model is built according to the strategy set out in the introduction. All attributes (except H^p) are used as inputs to the neural network with skip-layer connections. network weights are determined by minimizing the standard squared error loss function plus the sum of squared weights penalty term. The selection of the number of hidden units and the value of the weight decay parameter is based on 10-fold cross-validation. Neural network training and parameter selection is done on 80% of the data (randomly drawn), the remaining 20% is reserved for model evaluation. Multiple restarts (10) with different randomly selected initial weight vectors are performed to "ensure" a good locally optimal network solution.

Table 2. Model selection. The entries display R_{in}^2/R_{cv}^2

N_h	weight decay value λ					
	0.5	0.1	0.05	0.01	0.001	0
0	0.72/0.70	0.73/0.70	0.73/0.70	0.73/0.70	0.73/0.70	0.73/0.70
2	0.72/0.70	0.78/0.70	0.82/0.77	0.88/0.79	0.89/0.78	0.90/0.78
4	0.73/0.70	0.73/0.70	0.85/0.73	0.91/0.81	0.93/0.81	0.95/0.61
6	0.73/0.69	0.78/0.69	0.85/0.78	0.91/0.84	0.95/0.84	0.96/0.57

The intermediate results of the model selection process are displayed in table 2. The cells display the in-sample and out-of-sample coefficient of determination (R^2) for each neural network characterized by the network parameters N_h (number of hidden units) and λ (weight decay value). The in-sample R^2, which is denoted by R_{in}^2, represents the R^2 of the final neural network when fitted to the 80% of the data. The out-of-sample R^2 which is denoted by R_{cv}^2, is calculated from the vector of predictions obtained during the cross-validation procedure (also on the same 80% of the data).

The neural network model with the highest R_{cv}^2 is selected as the final network model to be used for prediction purposes. From table 2 it can be derived that the best network consists of 6 hidden units and uses a weigh decay value of 0.001 during weight estimation. According to the R_{cv}^2 criterion this neural network model improves over the parametric model found after manually transforming some of the variables.

The remaining 20% of the data (106 observations), which were randomly selected from the total sample, are used to assess the out-of-sample and in-sample R^2 achieved by the neural network. A comparison with the parametric models M_1 and M_2 is given in table 3.

Table 3. R^2 of the neural network, M_1, and M_2

Model	in-sample R^2	out-of-sample R^2
M_1	0.73	0.77
M_2	0.80	0.86
neural network (6/0.001)	0.95	0.90

The neural network model automatically finds an approximation to the relationship that clearly improves over the fit of the simple linear model and even over the fit of the model used in (Belsley *et al*., 1980); both in-sample and out-of-sample.

2.4 Monotonicity index

It is often said that it is difficult to interpret the nonlinear effects of the neural network model. From a practical point of view, it is interesting to know, whether the neural network output behaves monotonically with respect to one or more explanatory variables. This knowledge contributes to our understanding of the model. For this purpose, we have analyzed the behavior of the partial derivatives of the output with respect to the input variables as follows. For every explanatory variable we compute the partial derivative df/dx_i at each data point x_p. Here f denotes the neural network solution. The *degree of monotonicity* in x_i is defined as

$$\text{mon}(x_i) = \frac{1}{n}\left|\sum_{p=1}^{n} I^{+}\left(\frac{\partial f}{\partial x_i}(\mathbf{x}_p)\right) - I^{-}\left(\frac{\partial f}{\partial x_i}(\mathbf{x}_p)\right)\right|, \qquad \textbf{(3)}$$

where $I^+(z) = 1$ if $z > 0$ and $I^+(z) = 0$ if $z \leq 0$ and $I^-(z) = 1$ if $z \leq 0$ and $I^-(z) = 0$ if $z > 0$. n is the number of observations, and x_p is the p^{th} observation (vector). $0 \leq \text{mon}(x) \leq 1$. The degree of monotonicity of each of the explanatory variables is presented in table 4.

Table 4. Monotonicity of the house pricing model.

variable	$\text{mon}(x_i)$	variable	$\text{mon}(x_i)$
CRIM	0.62	AGE	0.87
ZN	0.34	DIS	0.99
INDUS	0.06	RAD	0.97
CHAS	0.13	TAX	0.98
NOX	1.00	PTRATIO	0.98
RM	0.76	LSTAT	0.76

A value of this index close to zero indicates a non-monotonic relationship, a value close to 1 indicates a monotonic relationship. Note that the dependence of the house price on the variables NOX, DIS, RAD, TAX and PTRATIO is (almost) monotonic. It is interesting to see that indeed some non-monotonic relationships are present. For example in the relation between INDUS and H^p, mon(INDUS) = 0.06. It turned out that above a certain quantity of non-retail business INDUS affects the house price positive, whereas below this level it effects the house price negatively. A possible reason for this could be the following. In areas with low business activity people are attracted by the pleasure of living, such as quietness, scenic environment, etc., which diminishes when the level of industry activity increases. House prices, consequently, are negatively affected by an increase in the level of industry. Living in an area with high business activity is attractive because commuting time is reduced to a minimum. When the level of business activity is increased in these areas, the area becomes even more attractive to live in. House prices, consequently, are positively affected by an increase in the level of industry.

3. Bond rating classification

3.1 Description of the case study

Bond ratings are subjective opinions on the ability to service interest and debt by economic entities such as industrial and financial companies, or municipals, and public utilities. Bond ratings are published by two major bond rating agencies, Moody's and Standard & Poor's, in the form of a letter code, ranging from AAA -for excellent financial strength-, to D for entities in default. Bond ratings are based on extensive financial analysis by the bond rating agencies. The exact determinants of a bond rating however are unknown, since the interpretation of financial information heavily relies on professional judgment.

During the last thirty years several attempts have been made to model corporate (industrial) bond ratings. The methods employed include linear regression, multiple discriminant analysis -linear and quadratic-, and neural networks. Linear regression models were proposed by (Horrigan, 1966), (Pogue and Soldofsky, 1969) and (West, 1970). (Pinches and Mingo, 1973), (Peavy, 1982), and (Belkaoui, 1980) employed discriminant analysis. (Moody, 1994), (Dutta and Shekhar, 1988) and (Kim *et al.*, 1993) recommended neural networks to model bond ratings. These studies were directed to general corporate bond ratings. In other studies (Altman and Katz, 1976) models of bond rating classification within specific industries are described.

3.2 Formulation of the empirical model.

The aim of the model as described here is to classify companies into the distinctive bond rating classes, based on their financial characteristics. Publications of bond rating agencies offer some perspicacity into the relevant factors that determine bond ratings. The bond rating analysis is directed to the following five main areas (cf. Hawkins, 1983):
-Profitability;
-Liquidity;
-Asset Protection;
-Indenture provisions;
-Quality of management.

The bond rating models use independent variables, often calculated as ratios, which are predominantly derived from public financial statements. However, not all of the above mentioned areas can be covered by financial statement figures. Aspects like quality of management, market positions and asset protection can only be captured to a limited extent. Also for indenture provisions, like subordination status, financial ratios are not applicable. Most of the ratios that are used in bond rating models can also be found in the literature on general financial statement analysis (cf. Lev, 1974).

3.3 Empirical study set up.

In the study setup, we follow the methodology described in (Moody, 1994). Two alternative approaches to model the classification of bond ratings are examined, a neural network with one hidden layer and a linear model. The neural network weights are determined by error-backpropagation, the coefficients in the linear model are estimated by OLS like in section 2.2. Comparing neural networks with multiple discriminant analysis would require a multiple output architecture of neural networks. Although neural networks are suitable for more than one output, we restrict our study to a one output architecture. Theoretically these architectures are equivalent, since every function can be approximated arbitrarily well by a neural network with one hidden layer.

From the Standard & Poor's Bond Guide (April 1994) 296 companies were selected. The bond ratings of these companies range from AAA to D. The ratings are not homogeneously distributed. The largest classes are A, BBB and B. Only very few selected companies have ratings lower than CCC. Therefore, we decided to remove all ratings below CCC. Like in other studies, the + and - signs were omitted (for example, AA+, AA, and AA- are all considered as AA). The bond ratings are quantified by assigning 0 to AAA until 6 to CCC.

From the S&P Bond Guide several financial figures have been obtained. From Datastream additional financial figures and ratios relating to leverage, coverage, liquidity, profitability, and size were downloaded. These figures have been restated to 5-year averages and trend indicators, resulting in 45 explanatory variables. For each variable the linear correlation with the quantified bond rating was calculated. Occasionally, a linear correlation test may not find possible non-linear correlation between input and output, although this will occur rarely in practice. The problem with measuring non-linear correlation is that it is highly arbitrary, since the nature of non-linearity between the variables is unknown.

It was found that neural networks trained on all 45 variables resulted in lower quality models than models based on only a selection of these variables. The four variables with the highest correlation are presented in table 5. The variables represent the level and stability of profits and the liquidity. The matrix of cross-correlations between all attributes is presented in table 9 in the appendix.

Table 5. Definition of the model variables

symbol	definition
Cov	3 years average interest coverage ratio
CF/D	5 years average cash flow to debt ratio
N.Pr.	5 years average net profit (in 100 millions)
D/C	5 years average debt to capital ratio

Several neural network architectures were evaluated varying the number of hidden neurons (2, 4, 8, 12, 20), the learning rate (0.1, 0.01), the momentum term (0.8, 0.1, 0.01), the type of activation function in the output layer (sigmoid, linear) and batch or online weight update. The performance of the neural network is expressed by prediction error (PE) as defined as:

$$PE = \frac{1}{|C|} \sum_{p \in C} (t^p - y^p)^2 \qquad \textbf{(4)}$$

based on a 5-fold cross validation. Here C denotes the validation set, t^p is the target pattern and y^p the actual outcome of the neural network.

The total set of patterns was divided into six mutually exclusive subsets, each containing (almost) 50 patterns. The relative distribution of the subsequent classes are approximately equal for all subsets. For each neural network architecture 5 different training runs were executed, each with another set serving as a test set. Training was accomplished on the remaining four subsets. During the training process, the performance was measured on both the training set and the test set. Training was stopped as soon as the lowest PE on the test set was reached.

The performance of the final network was tested on the training set, the test set, and the extra holdout sample. To compute the percentage of correct classification (PCC), the prediction of the neural network was rounded to the nearest discrete value (4/5 rounding). This value was compared to the actual class value.

3.4 Results

The final results show only minor differences for the evaluated architectures. Both the PE and the PCC were rather stable: PEs are in the range between 0.470 and 0.500, and the PCC varies between 46% and 60% (see table 7).
It is remarkable that although the PEs are rather high, PCCs are slightly better than those of the (linear) models as described in earlier studies. This can be explained by the fact that the PCC only refers to errors smaller than 0.5. Errors larger than 0.5 receive equal weights, since both large and medium errors are considered to be false classifications, regardless of the magnitude of the error. Nevertheless, in former studies the PCC is regarded as the main performance indicator. However, the PCC is not suitable as a error function in backpropagation neural networks, since such a function does not have a contineous first derivative.

An important result is that the architecture can be kept fairly simple. A neural network with 8 hidden neurons, sigmoid squashing functions, a learning rate of 0.1 and a momentum of 0.1 has a cross-validation error of 0.522 (PE) and a PCC of 51%. The results on the training set do not differ from the results on the test set and the extra holdout sample, indicating that overfitting in this case does not occur. When the number of instances in the training set was limited to 50, the PE dropped to 0.111 and the PCC consequently rose to 90%. The performance on the holdout sample however, was very bad (PE: 1.455, PCC: 32%), clearly showing the effects of overfitting.

To evaluate the results of our model with respect to earlier studies, we also employed linear regression models. These models are based on the same data set as used for the neural network model. Also for this regression analysis the 5-fold cross-validation method was implemented, resulting in 5 equations, that were tested on 5 holdout samples (see table 12 in the appendix).

The coefficient of the input net-profit is rather volatile, although its correlation to the output is moderate. The signs of the coefficients correspond to what is expected on the ground of economic plausibility. The t-statistics regarding Net Profit (N.Pr.) are rather low (< 1.96), indicating that this factor is not very appropriate in a linear model. However, this does not imply that it is not judicious to utilize this variable in a neural network model, since the statistics are only valid for linear models. Exclusion of the variable Net Profit from the linear model resulted in a slightly worse performance.

Table 6. PCC calculated using 5-fold cross validation

CV set	data set		holdout sample	
	neural network	linear	neural network	linear
1	47%	47%	54%	51%
2	51%	51%	46%	31%
3	51%	49%	46%	53%
4	48%	46%	52%	49%
5	50%	47%	57%	51%
	49%	48%	51%	47%

For each data set and corresponding holdout sample the percentage of correct classification was calculated. The results are presented in table 6. It is remarkable that in 4 of 5 cases the performance on the holdout sample is better than on the data set. This is caused by the presence of "difficult" (i.e. contradicting, or strongly differing) patterns in subset 2. For these patterns the error rate will be much higher compared to the remaining patterns. If these patterns are included in the training set (subsets 1, 3, 4, and 5), the PE is relatively high. This holds for the neural network and for the linear model as well. The overall out-of-sample performance of the simple linear model (47%) is slightly lower than the neural network models (51%).
The results show clearly the advantage of applying cross-validation. If training was performed using a single holdout sample, the values of the PCC would fluctuate between 46% and 57%, depending on whether subset 2, 3, or 5 was chosen as the holdout sample.

3.5 Monotonicity index

Like in section 2.4, we compute the monotonicity index with respect to the explanatory variables. The results are presented in table 7.

Table 7. Monotonicity index of the bond rating model

variable	mon(x_i)
D/C	1.00
Cov	0.94
CF/D	1.00
N.Pr	0.78

The monotonicity of D/C and CF/D agrees to the general theories on financial statement analysis. The more debt is covered by excess asset value or operating cash flow, the lower is the risk of insufficient available cash to service debt. The non-monotonicity of N.Pr may be due to dual nature of this variable, since it measures both firm size as profitability. The literature on bond rating is not decisive whether size can be regarded as debt protective. Intuitively, large firms may benefit from external financial support, for example, from the government or unions, since their failure may cause distress to other firms or society. On the other hand, failure of large firms did actually occur in the past. Although a large loss amount would deteriorate debt protection from a debt coverage perspective, it also indicates that the firm is large, and may benefit from external support. This size element is not applicable to Cov, since this is a ratio variable. The degree of nonmonotonicity of Cov may be explained by the higher volatility that usually accompanies high interest coverage. Bond rating analysts favor stable coverage ratios over high coverage. A firm may be highly profitable since it operates in growing markets. However, since growing markets will eventually mature, uncertainties may arise as to whether the firm is capable of adjusting to market changes.

4. Software.

In our experiments we have used the statistical package SPLUS for UNIX and the neural network simulation software Neuroshell2. SPLUS provides an interactive computing environment for graphical data analysis, statistics and computational programming. Neural networks are constructed in SPLUS using the S-function developed by Ripley (1993). This code is public available by anonymous ftp from markov.stats.ox.ac.uk (192.76.20.1) in the directory pub/S. It implements a standard feed-forward neural network with one hidden layer, no recurrent connections, and one output unit; the squashing function of the hidden units are sigmoid, the squashing function of the output unit can be sigmoid or linear. Skip layer connections from input to output can be provided. Estimation of the weights is done by a quasi-Newton general purpose optimizer, where the first derivatives are calculated by back-propagation.

Neuroshell2 is a neural network simulation package running on Windows PCs. It has facilities for designing neural networks, modules for data-processing and several different training algorithms. Several activation functions are supported, linear, logistic and Gaussian among them.

5. Conclusions & future research

We applied standard statistical techniques and neural networks to two economic classification problems. In particular we addressed different aspects of neural network modelling that are essential to obtain reliable predictions. The flexibility and the degree of non-linearity of the network is optimized by choosing optimal combinations of the weight decay parameter and the number of hidden neurons.

In the first case study the house price in Boston is estimated by linear regression, a modified linear model and a neural network architecture. It is shown that the modified linear model improves the linear model but the neural network has a significant lower error.

In the second case a model was developed to classify companies in different bond rating classes, using again a linear and neural network approach. It was found that the neural network slightly outperformed the linear model on the percentage of correct classifications (51% versus 47%). These results are in line with previous observations of other authors. Sensitivity analysis revealed that the neural network classifier clearly shows non-linear behavior in the data set. It was shown that the derivative of the classifier with respect to the interest coverage has different signs in the domain of the data set. A potential advantage of neural networks to provide insight of the confidence of the model is not supported by our findings.

In concluding, neural networks can be a valuable tool for either classification or prediction, especially when no parametric model is known or can only be developed at high costs. It has been noted that very often economists in banks do not agree on the parametric model, which is also a strong argument for using neural networks and letting the data decide. Since neural networks use very limited a priori knowledge the quality of the data is extremely important. If the data set is small neural network techniques are likely to fail because one is easily trapped in overfitting or finding local optima.

In both cases it was shown that the response function behaves monotonic with respect to several input variables. This is not surprising since one expects that most classification problems in economics and accounting possess monotonicity properties (Farley and Lin, 1990, Berndsen and Daniels, 1994). It is therefore natural to impose monotonicity constraints on neural network architecture because it reflects the generic properties of the underlying domain. This approach is particularly useful if the neural network tends to overfit the data by "spurious" oscillations near the classification boundary. This behavior typically occurs if the number of hidden neurons is larger. These spurious oscillations can be suppressed by addition of monotonicity constraints. The implementation of monotonicity constraints in the neural network training algorithm has been studied in (Archer and Wang, 1993) and (Wang, 1994). Currently, we study a certain class of monotonic neural networks that are inherently monotonic on a subset of the input variables. It can be shown that any monotonic function of n-variables can be approximated arbitrarily well by neural networks of this type. Algorithms for training of these special type of neural networks are under study and have shown good performance on laboratory test data.

Acknowledgement

Part of the research presented in this paper was done within an EC-funded network of the SPES programme, contract number 0065, which the authors gratefully acknowledge. The SPES (Stimulation Plan for Economic Science) project, entitled "Artificial Intelligence approaches to modelling in Economics", is a joint research project with participants from Heriot-Watt University (United Kingdom), Tilburg University (The Netherlands), Politecnico Milano (Italy), ABN/AMRO Bank (The Netherlands), and Digital Equipment Europe (France).

References:

Altman, E.I. (1994): Corporate distress diagnosis: comparisons using linear discriminant analysis and neural networks, *Journal of banking and finance*, **18**, 505-529.

Altman, E. and Katz, S. (1976): Statistical bond rating classification using financial and accounting data. in: *Proceedings of the Conference on Topical Research in Accounting*, ed. Schiff, M. and Sorter, G.

Archer, N.P., Wang, S. (1993): Application of the back propagation neural network algorithm with monotonicity conditions for two-group classification problems, *Decision Sciences*, **24**, 60-75

Belkaoui, A. (1980) Industrial bond ratings: A New Look. *Financial Management*, **9**, 44-50

Belsley, D., Kuh, E., and Welsch, R. (1980): Regression diagnostics: Identifying influential data and sources of collinearity, John Wiley & Sons, New York

Berndsen, R., and Daniels, H. (1994): Causal Reasoning in Economic Systems, *Journal of Economics and Control*, **18**, 251-271

Boritz J.E., Kennedy, D.B. and Augusto de Miranda e Albuquerque (1995): Predicting corporate failure using a neural network approach, *Intelligent Systems in Accounting, Finance and Management*, **4**, 95-111

Dutta, S., and Shekhar, S (1988): Bond rating: A non-conservative Application of neural networks, *Proceedings of the IEEE Conference of San Diego*

Farley, A.M., and Lin, K.P. (1990): Qualitative Reasoning in Economics, *Journal of Economic Dynamics and Control*, **14**, 465-490

Feelders, A, and Verkooijen, W. (1996): On the Statistical Comparison of Inductive Learning Methods, in: *Learning from Data* (Fisher, D and Lenz, H.J. (Eds), Springer

Geman, S., Bienenstock, E, and Doursat, R. (1981): Neural networks and the bias/variance dilemma, *Neural Computation*, **4**, 817-23

Harrison, D. and Rubinfeld, D. (1978): Hedonic prices and the demand for clean air, *Journal of Environmental Economics and Management*, **53**, 81-102.

Hawkins, D.F. (1983): Rating industrial bonds, Financial Executives Research Foundation, Morristown, N.J.

Horrigan, J.O. (1966): The determination of long-term credit standing with financial ratios, *Journal of Accounting Research*, **4**, supplement, 44-62

Janssen, J. (1992): *De prijsvorming van bestaande koopwoningen*, PhD thesis, Catholic University Nijmegen

Kim, J.W., Weistroffer, H.R., Redmond, R.T. (1993): Expert systems for bond rating: a comparative analysis of statistical, rule-based and neural network systems, *Expert Systems*, **10**, 167-188

Moody, J. (1994): Architecture selection strategies for neural networks: application to corporate bond rating prediction, *Neural networks in the capital market*, John Wiley, New York.

Lev, B (1974): Financial statement analysis: a new approach, Englewood Cliffs, London

Peavy, J.W. (1982): Long run implications of industrial bond ratings as risk surrogates, *Journal of Bank Research*, **34**, 331-341

Pinches, G.E. and Mingo, K.A. (1973): A multivariate analysis of industrial bond ratings, *The journal of Finance*, **28**, 1-17

Pogue, T.F. and Soldofsky, R.M. (1969): What's in a bond rating?, *Journal of financial and quantitative analysis*, **4**, 201-228

Ripley, B.D. (1993): Flexible non-linear approaches to classification, in: *From Statistics to Neural Networks; Theory and Pattern Recognition Application*s, Springer-Verlag

Stone, M. (1974): Cross-validatory choice and assessment of statistical predictions, *Journal of the Royal Statistical Society*, **36**: 111-147

Verkooijen, W.J.H. (1996): Neural Networks in Economic Modelling, An Empirical Study, CentER dissertation, ISBN 90-5668-00-010-2.

Wang, S. (1994): Neural network techniques for monotonic nonlinear models, *Computer & Operations Research*, **21,** 143-154

West, R.R. (1970): An alternative approach to predicting corporate bond ratings, *Journal of Accounting Research*, **7**, 118-125.

Appendix

Table 8. All pairwise cross-correlations of Boston housing case.

	CR	ZN	IND	CH	NO	RM	AG	DIS	RAD	TAX	PT	LS
CRIM	1.0	-0.2	0.4	-0.1	0.4	-0.2	0.4	-0.4	0.6	0.6	0.3	0.5
ZN	-0.2	1.0	-0.5	-0.0	-0.5	0.3	-0.6	0.7	-0.3	-0.3	-0.4	-0.4
INDUS	0.4	-0.5	1.0	0.1	-0.8	-0.4	0.6	-0.7	0.6	0.7	0.4	0.6
CHAS	-0.1	-0.0	0.1	1.0	0.1	0.1	0.1	-0.1	-0.0	-0.0	-0.1	-0.1
NOX	0.4	-0.5	0.8	0.1	1.0	-0.3	0.7	-0.8	0.6	0.7	0.2	0.6
RM	-0.2	0.3	-0.4	0.1	-0.3	1.0	-0.2	0.2	-0.2	-0.3	-0.4	-0.6
AGE	0.4	-0.6	0.6	0.1	0.7	-0.2	1.0	-0.7	0.5	0.5	0.3	0.6
DIS	-0.4	0.7	-0.7	-0.1	-0.8	0.2	-0.7	1.0	-0.5	-0.5	-0.2	-0.5
RAD	0.6	-0.3	0.6	-0.0	0.6	-0.2	0.5	-0.5	1.0	0.9	0.5	0.5
TAX	0.6	-0.3	0.7	-0.0	0.7	-0.3	0.5	-0.5	0.9	1.0	0.5	0.5
PTRATIO	0.3	-0.4	0.4	-0.1	0.2	-0.4	0.3	-0.2	0.5	0.5	1.0	0.4
LSTAT	0.5	-0.4	0.6	-0.1	0.6	-0.6	0.6	-0.5	0.5	0.5	0.4	1.0
H^P	-0.4	0.4	-0.5	0.2	-0.4	0.7	-0.4	0.2	-0.4	-0.5	-0.5	-0.7

Table 9. All pairwise cross-correlations of the Bond ratin case.

	Cov	CF/D	N.Pr	D/C
Cov	1.0	0.7	0.5	-0.6
CF/D	0.7	1.0	0.5	-0.4
N.Pr	0.5	0.5	1.0	-0.3
D/C	-0.6	-0.4	-0.3	1.0
rating	-0.7	-0.6	-0.6	0.5

Table 10. The OLS estimates (with standard errors) of modelling MEDV by M_1

attribute	value	st. error	*t*-value
(intercept)	36.5	5.10	7.14
CRIM	-0.11	0.033	-3.29
ZN	0.046	0.014	3.38
CHAS	2.69	0.86	3.12
NOX	-17.77	3.82	-4.65
RM	3.81	0.42	9.12
DIS	-1.48	0.20	-7.40
RAD	0.31	0.066	4.61
TAX	-0.012	0.0038	-3.28
PTRATIO	-0.95	0.13	7.28
LSTAT	-0.52	0.051	-10.35
R^2	0.74		

Table 11. The OLS estimates (with standard errors) of modelling log(MEDV) by M_2.

attribute	value	st. error	*t*-value
(intercept)	9.76	0.15	65.22
CRIM	-0.012	0.0012	-9.53
CHAS	0.091	0.033	2.75
NOX^2	-0.0064	0.0011	-5.64
RM^2	0.0063	0.0013	4.82
log(DIS)	-0.19	0.033	-5.73
log(RAD)	0.096	0.019	5.00
TAX	-0.00042	0.00012	-3.43
PTRATIO	-0.031	0.0050	-6.21
log(LSTAT)	-0.37	0.025	-14.84
R^2	0.81		

Table 12. The linear bond rating model

Variable	CV set	coefficient	st. error	t-value
D/C	1	0.0093	0.002799	3.3322
	2	0.0111	0.002600	4.2699
	3	0.0101	0.003060	3.3003
	4	0.0110	0.002721	4.0420
	5	0.0145	0.003140	4.6168
Cov	1	-0.617	0.120171	5.1335
	2	-0.595	0.116766	5.0968
	3	-0.633	0.122412	5.1706
	4	-0.589	0.131419	4.4823
	5	-0.605	0.124755	4.8497
CF/D	1	-0.991	0.22021	4.5005
	2	-0.955	0.19100	5.0000
	3	-0.683	0.21094	3.2383
	4	-0.958	0.21181	4.5228
	5	-0.910	0.19000	4.7893
N.Pr.	1	-2.796	1.736507	1.6100
	2	-2.018	1.650568	1.2206
	3	-4.549	1.776569	2.5608
	4	-3.054	1.858843	1.6428
	5	-1.561	1.782571	0.8757
Intercept	1	3.379		
	2	3.163		
	3	3.451		
	4	3.250		
	5	2.982		