

Case Study

Symbolic rule extraction from neural networks: An application to identifying organizations adopting IT

Rudy Setiono^{a*}, James Y.L. Thong^b and Chee-Sing Yap^a

^a*Department of Information Systems and Computer Science, National University of Singapore, Kent Ridge, Singapore 119260*

^b*Department of Information and Systems Management, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*

Abstract

Interest in the application of neural networks as tools for decision support has been growing in recent years. A major drawback often associated with neural networks is the difficulty in understanding the knowledge represented by a trained network. This paper describes an approach that can extract symbolic rules from neural networks. We illustrate how the approach successfully extracted rules from a data set collected from a survey of the service sectors in the United Kingdom. The extracted rules were then used to distinguish between organizations using computers from those that do not. The classification scheme based on these rules was used to identify specific segments of a market for promoting adoption of information technology. The extracted rules are not only concise but also outperform discriminant analysis in terms of predictive accuracy.

Keywords. Backpropagation algorithm; neural networks; symbolic rules; IT adoption

Original submittal: Sept 6, 1996.

First response: July 24, 1997.

Resubmittal: Sept 30, 1997.

Accepted: April 16, 1998.

*Corresponding author

1. Introduction

Successful applications of neural network techniques in business research have been widely reported in the literature. Neural networks have been used to predict bank failures [20], to perform bond rating [4], to analyse financial data [2], to assess the financial health of savings and loan associations [15], to select forecasting models [19], and to assess the need for end users in the planning of the development part of information systems [8]. The most attractive feature of neural networks that leads to their wide degree of user acceptance is their accuracy in predicting results. For some applications, such as bank failures, it was reported that neural networks achieve better predictive accuracy than discriminant analysis, logistic regression, k-nearest neighbor, or the decision tree method ID3.

A major drawback of neural networks as tools for predicting is the lack of their ‘explanation’ capability when trained: it is generally difficult for a user to understand *how* a network arrives at a particular conclusion because of its complex architecture. However, it is often desirable to have a set of meaningful and coherent rules that describes why a pattern is classified as a member of a certain class. Such rules are a form of knowledge that can be verified by human experts, and then passed on and expanded. Scientific discovery may then occur.

The search for interesting and useful patterns in a data set has spawned the interdisciplinary research area known as data mining, which generally applies some machine learning techniques to achieve its goal of discovering the patterns hidden in the data. Some of the machine learning techniques employed in data mining include inductive methods that build decision trees such as C4.5 [12] and CART [1] and clustering algorithms such as the k-nearest neighbor [3] and COBWEB [5]. The use of neural networks as tools for data mining has been largely limited, due to the lack of an explanation capability. Recent developments in algorithms that extract rules from neural networks, however, have made the neural network approach a useful and viable technique in data mining [9].

The objectives of this paper are twofold: (1) to provide a brief description of NeuroRule, a system that we have developed to extract decision rules from neural networks [17], and (2) to illustrate its application as a tool in business research. The application chosen is to distinguish characteristics of organizations using computers. The data for this example was collected from 638 business organizations in the service sectors in the United Kingdom through a mail survey [23].

2. Neural networks and rule extraction

[Figure 1 HERE]

A neural network simulates the operations of a human brain. A typical neural network is presented in Figure 1. The nodes or units in the network are organized into three layers: input, hidden, and output. The input nodes take in the values of the independent variables while the output nodes generate the predicted values of the dependent variables. There are connections between all the nodes in adjacent layers. There is a separate connection from each input node to each hidden node and from each hidden node to each output node. Each connection has a connection strength or weight. Each node in the hidden layer or output layer receives inputs from other nodes in the preceding layer. This node's activation value is computed by applying a nonlinear activation function to the total weighted input coming into the node (see Figure 2). Two commonly used functions are sigmoidal:

$$f(x) = 1/(1 + \exp(-x))$$

or hyperbolic tangent:

$$f(x) = (\exp(x) - \exp(-x))/(\exp(x) + \exp(-x))$$

[Figure 2 HERE]

The network “learns” by adjusting the interconnection weights between nodes. Given a pattern (x, y) , where x is the input and y is its target value, the predicted output is computed by passing the values of x through the input nodes, computing its activation values at the nodes in the hidden layer, and finally computing the activation values at the output nodes. The output unit values represent the predicted value for input x . Error is computed as the difference between the predicted value and the actual value. It is backpropagated through the network and is used to adjust the connection weights. The weights are adjusted so that the sum of squared errors of all patterns is minimized. The network is trained when a set of weights that minimizes the sum of squared error function is obtained. Interested readers may refer to references [10, 13] for a more detailed discussion.

In NeuroRule, the process of rule extraction from a neural network can be summarized as:

1. Select and train a network to meet the prespecified accuracy requirement.
2. Remove the redundant connections in the network by pruning while maintaining its accuracy.
3. Discretize the activation values in the hidden nodes of the pruned network by clustering.
4. Extract rules that describe the network outputs in terms of the discretized activation values of the hidden nodes.
5. Generate rules that describe the discretized activation values of the hidden nodes in terms of the network inputs.
6. Merge the two sets of rules generated in Steps 4 and 5 to obtain a set of rules that relates the inputs and outputs of the network.

In Step 1, a suitable network is selected and trained for the problem on hand. This network can be trained by either the backpropagation algorithm [14] or the Newton-Raphson algorithm [11].

The process of rule extraction from neural networks can be made substantially easier after the complexity of the network is reduced. Steps 2 and 3 reduce the network complexity and facilitate the process of rule extraction by pruning redundant connections of the network and clustering the activation values of the hidden nodes. Network pruning, though apparently complex, is really quite simple in concept; it involves computations that identify which connections in the network can be removed without changing the accuracy of the prediction. Next, we cluster the activation values of the nodes in the hidden layer. This again is intended to reduce the complexity of the network while maintaining the predictive accuracy of the network. Thus, the reduction in the number of connections and number of activation values of the nodes in the hidden layer will make it easier to produce a set of compact rules for the data. Those interested in the case study can skip the rest of this section, moving straight to the example.

2.1 Neural network training and pruning

Step 1 of NeuroRule involves training of a fully connected neural network. Given a set of input-output pairs, the training of the neural network entails finding a set of weights that minimizes the network error function. The error function that we minimize is the cross-entropy function. Experimental results have indicated that the cross-entropy error function enjoys higher convergence than the normal sum of squared error function [22].

One of the difficulties with using feedforward neural networks is the need to determine the optimal number of hidden nodes before the training process can begin. Too many hidden nodes may lead to overfitting of the data and poor generalization, while too few hidden nodes may not produce a network that learns the data well. To overcome this difficulty, researchers usually start with more than the necessary hidden nodes. Redundant nodes are subsequently eliminated from the network after the training pro-

cess has terminated. In order to eliminate irrelevant input nodes and hidden nodes, a trained network needs to be trimmed by pruning. The pruning algorithm must be able to find those connections in the network that are redundant. Removing the redundant connections from the network usually increases the capability of the network to generalize; i.e., to predict new patterns not used for training with a satisfactory accuracy rate. An input node that has all its connections removed can be eliminated without affecting the network accuracy. Similarly, a hidden node that has all its input or output connections eliminated should also be removed from the network. Step 2 of NeuroRule prunes redundant connections and removes unneeded input and hidden units..

There have been many algorithms for neural network pruning proposed in the past few years [6, 21]. A weight-decay term is normally added to the error function so that connections that are redundant will have weights with small magnitude at the end of training. Network connections are removed based on their magnitude. The details of the pruning algorithm that makes use of the weight-decay term can be found in [16]. This pruning algorithm has been shown to be very successful in finding minimal networks for the problems tested.

2.2 Clustering activation values of hidden nodes

Step 3 of NeuroRule performs clustering or discretization of activation values in the hidden nodes of samples that are correctly classified by the pruned network. Clustering is accomplished in NeuroRule by a simple greedy clustering algorithm (GCA).

When the hyperbolic tangent function is used as the activation function, the activation value of a hidden node can take any value in the interval $[-1, 1]$. The clustering algorithm tries to merge as many activation values of a hidden node as possible into a single value as long as the merging does not produce conflicting data; i.e. there are two or more identical discretized activation values for input patterns that belong to different expected outputs. After sorting the activation values at the hidden node, pairs of adjacent values are selected for possible merging based on their distance. The pair of distinct adjacent

values with the shortest distance is considered first. If possible, they will be merged by replacing the larger value with the smaller one. Otherwise, a pair with the next shortest distance will be considered. This process is repeated until no further pair of values can be merged.

2.3 Rule extraction from networks with discretized activation values

Step 4 of NeuroRule generates rules which relate the discretized activation values in the hidden nodes with the expected output values in the connected output nodes. Step 5 generates rules that relate the activation values in the hidden nodes with the attribute values at the connected input nodes. When the number of clusters in the hidden nodes is small, it is easy to obtain rules to describe the network outputs in terms of the activation values. Similarly, when the number of inputs connected to a hidden node is small, it will be trivial to extract rules that describe how each cluster of activation values is obtained from the input values. A general purpose algorithm X2R [7] was implemented to automate the rule generation process. It takes as input a set of discrete input patterns with the expected output values and produces the rules describing the relationship between the patterns and their expected output values.

X2R produces perfect rules, i.e., it does not introduce misclassification. The clusters at the hidden nodes have been obtained using those activation values from the correctly classified samples only. As a result, the accuracy of the rules extracted from a network is at least as high as the accuracy of the network.

Each condition of the rules generated in Step 4 is given in terms of the discretized hidden node activation values. The rules generated in Step 5 explain how each of the discretized activation values is obtained in terms of the input attributes of the data. Merging of the two sets of rules is achieved by replacing the conditions of the rules from Step 4 by the inputs that describe each activation value obtained from Step 5.

3. Finding characteristics of organizations using computers

3.1 Data collection

The U.K. service industry was chosen for our case study because it is a part of the U.K. economy that has been undergoing rapid change as a direct result of the IT revolution. The service industry is divided into five sectors in the Standard Industrial Classification (SIC) for the U.K.: (1) transport and communication, (2) wholesale distribution, (3) retail distribution, (4) business and financial services, and (5) miscellaneous services including hotels and catering, repairs, and recreational services.

Data was collected from this industry through a mail survey. Before conducting this, the questionnaire was pre-tested in two phases. In the first, personal interviews were conducted with senior managers from six organizations. In the second phase, a questionnaire was mailed to 100 organizations selected at random from a directory provided by the Register of British Industry and Commerce. 78 responses were received. Based on the pre-testing, the questionnaire was slightly modified for clarity and presentation. The final questionnaire was mailed to a new random sample of 3000 organizations. 695 completed questionnaires were received. After eliminating organizations that were not in the service sectors, we obtained 638 usable questionnaires, out of which 443 were from organizations using computers and the rest were not.

A survey of the existing literature identified a large number of potential organizational characteristics that may influence IT adoption. Of these, ten research variables (see Table 1) were included, based on two main criteria: (1) they had been found in previous studies to be relevant to organizational use of computers, and (2) there must be a simple yet meaningful operational measure that could be incorporated in the mail questionnaire.

[Table 1 HERE]

One of the ten variables, PINFO, is real-valued. The remaining nine have discrete

integer values. The possible range for these values is from 1 to 5, except for TURNOVER, which has a possible range from 1 to 3. Before neural network training can start, the raw data need to be preprocessed. PINFO is normalized into the interval $[0, 1]$ by dividing all values by the largest value in the data, which is 0.92. There is only one nominal discrete variable which is SECTOR. Since this variable can have 1 of the 5 possible values from 1 to 5, five input nodes are used. The transport, wholesale, retail, banking and finance, and miscellaneous sectors are coded as $(1, 0, 0, 0, 0)$, $(0, 1, 0, 0, 0)$, $(0, 0, 1, 0, 0)$, $(0, 0, 0, 1, 0)$, $(0, 0, 0, 0, 1)$, respectively. The ordinal variables are coded using the *thermometer* coding scheme [18]. In this scheme, a variable with N possible values is also represented by N input nodes. For example, the variable TURNOVER has 3 possible values: 1 (indicating turnover of less than 1 million £), 2 (indicating turnover between 1 million £ and 5 millions £), or 3 (indicating turnover greater than 5 millions £), which are coded as $(0, 0, 1)$, $(0, 1, 1)$, and $(1, 1, 1)$, respectively. Under this scheme, the first coded input is 0 if and only if the turnover is less than 5 millions £ and the second coded input is 0 if and only if the turnover is less than 1 million £. It is possible to code the possible values of TURNOVER by using only 2 binary inputs: $(0, 0)$, $(0, 1)$, and $(1, 1)$. However, an advantage of the thermometer scheme is that the extra input connections from the additional input node to the nodes in the hidden layer increase the number of connection weights in the network and allow the network higher capacity to fit the data.

The network inputs required for all variables are summarized in Table 2. The table shows how the original 10 variables are assigned 44 input nodes in the network. Associated with each of the samples is an expected output value, which is either 1 or 0 depending on whether the organization is or is not using computers respectively. An organization using computers is correctly classified if the network output is greater than or equal to 0.50, while an organization not using computer is correctly classified if the network output is less than 0.50.

[Table 2 HERE]

3.2 Results from neural network training and pruning

[Table 3 HERE]

To test the robustness of the training and pruning algorithms, researchers usually train a large number of networks. In our case, 50 neural networks were trained. Each of these networks has 44 input nodes, 6 hidden nodes and 1 output node. The initial weights of the network connections were generated randomly and uniformly in the interval $[-1, 1]$.

The 638 samples were randomly divided into three sets. The training set consists of 60 % of the total (382 samples), the cross-validation set consists of 20 % of the total (128 samples), and the testing set consists of the remaining samples. There is a trade-off between the accuracy of the pruned network on the training samples and its complexity. More connections can be removed at the cost of reducing the accuracy of the resulting network. The cross-validation set is used to determine when the pruning process should be terminated. We continued pruning a network as long as its accuracy on the training samples was still at least 75 % and its accuracy on the cross-validation samples was within 10 % of the training accuracy. The results are summarized in Table 3. Note the increase in the average predictive accuracy (accuracy on the testing set) and the dramatic decrease in the number of connections after pruning.

3.3 Extracting rules from pruned networks

After pruning 50 fully connected networks having 6 hidden nodes, pruned networks with varying architectures were obtained. The following two examples are chosen to illustrate how rules are extracted by NeuroRule from networks having different number of hidden

nodes and connections. The network in the first example is chosen because it has the fewest connections among the 50 pruned networks. The network in the second example has the highest predictive accuracy.

Example 1.

[Figure 3 HERE]

One of the networks with the fewest connections left after pruning is shown in Figure 3. The accuracy rates of this network on the training (including cross-validation) and testing data sets are 77.06% and 78.91%, respectively. In this network, only inputs \mathcal{I}_2 and \mathcal{I}_7 remain. The activation values at the only remaining hidden node range from -0.83 to 0.24. The value of -0.83 is obtained when both \mathcal{I}_2 and \mathcal{I}_7 are equal to 1, while the value of 0.24 is obtained when the 2 inputs are zero. These values have been computed using the hyperbolic function. For $(\mathcal{I}_2, \mathcal{I}_7) = (1, 1)$, the activation is $f(-0.59\mathcal{I}_2 - 0.83\mathcal{I}_7 + 0.24) = -0.83$. For $(\mathcal{I}_2, \mathcal{I}_7) = (0, 0)$, the activation is $f(-0.59\mathcal{I}_2 - 0.83\mathcal{I}_7 + 0.24) = 0.24$.

Step 3 of NeuroRule divides the activation values into two clusters. All activation values that are less than 0.24 are grouped into the first cluster, i.e., each of these values is replaced by -0.83. The second cluster consists of only 1 activation value, 0.24. Of the 393 training samples that are correctly classified by the network, 310 samples have their hidden node activation values in the first cluster and the remaining 83 in the second cluster. All samples with activation values of -0.83 are from organizations using computers, while those with activation values of 0.24 are from organizations not using computers.

Given the two possible clusters of activation values at the hidden node, it is not difficult to obtain the rule that describes how each of these values are obtained since they are determined by only three inputs. We print inputs 2 and 7 of all correctly classified training samples and labeled each of these samples 1 or 2, depending on whether its activation values are grouped in the first or second cluster. The two binary inputs can produce a total of four unique combinations. These four patterns and their expected

outputs are used as input to step 4 of NeuroRule. A very simple rule is generated: if $\mathcal{I}_2 = \mathcal{I}_7 = 0$, then activation value = 0.24 (ie. organization is not using computers), otherwise activation value = -0.83 (i.e. organization is using computers). The coding scheme for the input data allows us to easily arrive at the rule in terms of the original attributes:

Rule 1:

- If $\text{TURNOVER} = 1$ and SECTOR is not Finance, then *not using computer*.
- Default rule. *Using computer*.

The overall accuracy rate of this rule and the rates for individual sectors are summarized in Table 4.

[Table 4 HERE]

Example 2.

[Figure 4 HERE]

A pruned network that has an average number of connections of 10 is shown in Figure 4. The accuracy rates of this network on the training (including cross-validation) and testing data sets are 78.04% and 82.81%, respectively. There are three hidden nodes in the network. NeuroRule found that there are three, three, and two clusters of activation values, respectively at the three hidden nodes.

We let α_i represent the cluster in which the activation value of hidden node i of a sample fall. Hence, $\alpha_1 = 1, 2, \text{ or } 3$, $\alpha_2 = 1, 2, \text{ or } 3$, and $\alpha_3 = 1 \text{ or } 2$ denote the cluster at the hidden nodes 1, 2, and 3 respectively. The rules extracted by NeuroRule in terms of the clustered activation values are as follows:

Rule 2(a):

If $\alpha_2 = 1$ and $\alpha_3 = 2$ OR

if $\alpha_1 = 2$ and $\alpha_2 = 1$ and $\alpha_3 = 1$ OR

if $\alpha_1 = 3$ and $\alpha_2 = 2$ and $\alpha_3 = 2$, then *not using computer*.

Default rule. *Using computer*.

Next, NeuroRule generates a set of rules that describes the activation values of the hidden nodes in terms of the inputs, and the set of rules that describe the outputs in terms of the activation values of the hidden nodes. Substituting the second set of rules into the conditions of the first set of rules, we obtain

Rule 2(b):

If $\mathcal{I}_1 = \mathcal{I}_7 = \mathcal{I}_{14} = \mathcal{I}_{28} = 0$ OR

if $\mathcal{I}_2 = \mathcal{I}_7 = \mathcal{I}_{14} = 0$ and $\mathcal{I}_{28} = 1$ OR

if $\mathcal{I}_2 = \mathcal{I}_7 = \mathcal{I}_{28} = 0$ and $\mathcal{I}_{14} = 1$, then *not using computer*.

Default rule. *Using computer*.

In terms of the original attributes of the data, we have the following equivalent set of rules:

Rule 2:

- If $\text{TURNOVER} < 3$ and SECTOR is not Finance and $\text{GROW} < 5$ and $\text{FORMAL} = 1$
OR
- if $\text{TURNOVER} = 1$ and SECTOR is not Finance and $\text{GROW} < 5$ and $\text{FORMAL} \geq 2$ OR
- if $\text{TURNOVER} = 1$ and SECTOR is not Finance and $\text{GROW} = 5$ and $\text{FORMAL} = 1$,
then *not using computer*.
- Default rule. *Using computer*.

The accuracy of this rule is summarized in Table 5.

[Table 5 HERE]

4. Discussion

For comparison purposes, we also analyzed the same data set using discriminant analysis, a method commonly applied for decision support. The results are summarized in Table 6. Note that the overall predictive accuracy rate of 76.56 % is lower than the rates of rules generated by NeuroRule. In fact, by knowing only the values of 2 input variables in Rule 1, it is possible to achieve a higher predictive accuracy than discriminant analysis.

Since the training network may terminate at different local minimum points when the networks are given different initial random weights, different sets of rules can be extracted from these networks. This nondeterministic characteristic of neural network training allows us to extract different sets of rules. The possibility of extracting different sets of rules is an advantage of using neural networks. One set of rules may be more useful than another, for instance, the first set of rules involves only those variables whose values are less costly to gather in the future.

[Table 6 HERE]

In deciding which rules should be applied to distinguish between organizations using computers and those that are not using them, two factors must be taken into consideration. These factors are simplicity of the rules and predictive accuracy of the rules. Simple rules can be extracted from sparse networks. However, their accuracy rates may not be as high as the accuracy of the rules extracted from networks with more connections. Extracting rules from denser networks can be expected to produce more complicated rules. One must resolve the trade-off between the two factors in selecting the final rules for the specific application domain.

Rule 1 tells us that organizations with turnover of less than 1 million pounds and not in the financial sector are likely to be non computer users. The accuracy of this rule is close to 79 %. Higher accuracy rates can be obtained from networks having more connections. We show in Example 2, a relatively more complex network than in Example 1. The rules extracted from this network produce higher accuracy, at the cost of more rules and more conditions in each rule. By including two additional variables GROW and FORMAL, Rule 2 achieves 4 % higher accuracy than Rule 1.

There is no clear cut criterion for deciding between Rule 1 and Rule 2. Rule 1 is the simplest while Rule 2 has the highest accuracy. Much of the criteria for rule selection are subjective. One's knowledge of the problem domain will help in determining the relative importance of rule simplicity and rule accuracy. In the domain of computer usage by organizations in the service industry in the United Kingdom that we investigated, Rule 2 is more appropriate. Not only is its overall predictive accuracy higher than Rule 1, but its accuracy on every subsector is also either the same or higher. This higher accuracy is achieved by including only two more variables in the conditions of Rule 2.

5. Conclusion

Neural networks have been applied widely as tools for decision support. Due to their complex nonlinear mapping of the data, neural networks are often viewed as unfathomable black boxes. Previously, it was very difficult to articulate the decision process of a trained network. Having explicit symbolic rules that explain the network outcomes can be beneficial in many ways. Explicit rules make it possible for the decision process to be verified by human experts. They may also provide new insight into the problem domain by providing interesting patterns buried in the data.

We have described in this paper a system that extracts decision rules from neural networks. The system was illustrated through an application that identifies characteristics of organizations using computers in the service sectors of the U.K. Three factors enable us to extract compact set of rules from neural networks that classify the samples

with high accuracy. The first is an effective neural network pruning algorithm. We have been able to prune networks so that very few input nodes, hidden nodes, and connections remain. The second factor is an algorithm that clusters hidden node activation values of a pruned network. This algorithm allows us to consider only a small number of different values and still maintain the accuracy of the original network. The third factor is an algorithm that generates classification rules from small datasets having discrete inputs.

The rules extracted from the neural networks were shown to achieve higher predictive accuracy than discriminant analysis for the same example. They can be used to identify specific segments of the service industry for promoting adoption of information technology. The system to extract rules from neural networks presented here is general and can be applied to other business applications where neural networks have been shown to be a method of choice in classification and prediction.

References

- [1] Breiman, L., Friedman, J.H, Olshen, R.A. and Stone, C.J., *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA., 1984.
- [2] Coakley, J.R. and Brown, C.E., “Artificial neural networks applied to ratio analysis in the analytical review process”, *Intelligent Systems in Accounting, Finance and Management*, 2, 1993, pp. 19–39.
- [3] Cover, T.M. and Hart, P.E., “Nearest neighbor pattern classification”, *IEEE Transactions on Information Theory*, IT-13, 1967, pp. 21–27.
- [4] Dutta, S., Shekhar S. and Wong, W.Y., “Decision support in non-conservative domains: Generalization with neural networks”, *Decision Support Systems*, 11(5), June 1994, pp. 527–544.
- [5] Fisher, D.H., “Knowledge acquisition via incremental conceptual clustering”, *Machine Learning*, 2(7), 1987, pp. 139–172.

- [6] Karnin, E.D., “A simple procedure for pruning back-propagation trained neural networks”, *IEEE Transactions on Neural Networks*, 1(2), 1990, pp. 239–242.
- [7] Liu, H. and Tan, S.T., “X2R: A fast rule generator”, in Proceedings of IEEE International Conference on Systems, Man and Cybernetics, IEEE Press, New York, 1995, pp. 388–391.
- [8] Lodewyck, R.W. and Deng, P.S., “Experimentation with a back-propagation neural network - An application to planning end user system development”, *Information and Management*, 24(1), January 1993, pp. 1–8.
- [9] Lu, H., Setiono, R. and Liu, H., “Effective data mining using neural networks”, *IEEE Transactions on Knowledge and Data Engineering*, 8(6), December 1996, pp. 957–961.
- [10] Muller, B., Reinhardt, J. and Strickland, M.T., *Neural Networks: An Introduction*, Springer-Verlag, Berlin, 1990.
- [11] Piramuthu, S., Shaw, M.J. and Gentry, J.A., “A Classification approach using multi-layer neural-networks”, *Decision Support Systems*, 11(5), June 1994, pp. 509–526.
- [12] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, CA. 1993.
- [13] Rojas, R., *Neural Networks: A Systematic Introduction*, Springer-Verlag, Berlin, 1996.
- [14] Rumelhart, D.E. and McClelland, K. (Eds.), *Parallel Distributed Processing: Explorations in Microstructure of Cognition*, Vols. 1–2, MIT Press, 1986.
- [15] Salchenberger, L.M., Cinar, E.M. and Lash, N.A., “Neural networks: A new tool for predicting thrift failures”, *Decision Sciences*, 23(4), July/August 1992, pp. 899–916.
- [16] Setiono, R., “A penalty-function approach to pruning feedforward neural networks”, *Neural Computation*, 9(1), January 1997, pp. 185–204.

- [17] Setiono, R. and Liu, H., “Symbolic representation of neural networks”, *IEEE Computer*, March 1996, pp. 71–77.
- [18] Smith, M., *Neural Networks for Statistical Modeling*, Van Nostrand Reinhold, New York, 1993.
- [19] Sohl, J.E. and Venkatachalam, A.R., “A neural network approach to forecasting model selection”, *Information and Management*, 29(6), 1995, pp. 297-303.
- [20] Tam, K.Y. and Kiang, K.Y., “Managerial applications of neural networks: The case of bank failure predictions”, *Management Science*, 38(7), July 1992, pp. 926–947.
- [21] Thodberg, H.H., “Improving generalization of neural networks through pruning”, *International Journal of Neural Systems*, 1(4), 1991, pp. 317–326.
- [22] van Ooyen, A. and Nienhuis, B., “Improving the convergence of the backpropagation algorithm”, *Neural Networks*, 5(3), 1992, pp. 465–471.
- [23] Yap, C.S., “Distinguishing characteristics of organizations using computers”, *Information and Management*, 18(2), 1990, pp. 97–107.

Table 1: Variables included in the study

| Organizational Characteristics | Research Variables | Variable Labels |
|--------------------------------|-------------------------------------------------|-----------------|
| 1. Size | Annual sales turnover | TURNOVER |
| 2. Sector | Industrial sector | SECTOR |
| 3. Performance | Average return on capital over the last 3 years | ROCE |
| | Growth of business over the last 3 years | GROW |
| 4. Task | Routineness of work activities | ROUTINE |
| 5. People | Percentage of information workers | PINFO |
| 6. Structure | Degree of formalization of communications | FORMAL |
| | Degree of centralization of decision making | CENTRAL |
| 7. Environmental factors | Competitiveness of the market | COMPETE |
| | Predictability of customer's requirements | PREDICT |

Table 2: Research variables and their network inputs

| Variable | Type | Possible values | Input |
|----------|------------|-----------------|--------------------------------------------------------------------------------------------|
| TURNOVER | Ordinal | 1-3 | $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3$ |
| SECTOR | Nominal | 1-5 | $\mathcal{I}_4, \mathcal{I}_5, \mathcal{I}_6, \mathcal{I}_7, \mathcal{I}_8$ |
| ROCE | Ordinal | 1-5 | $\mathcal{I}_9, \mathcal{I}_{10}, \mathcal{I}_{11}, \mathcal{I}_{12}, \mathcal{I}_{13}$ |
| GROW | Ordinal | 1-5 | $\mathcal{I}_{14}, \mathcal{I}_{15}, \mathcal{I}_{16}, \mathcal{I}_{17}, \mathcal{I}_{18}$ |
| ROUTINE | Ordinal | 1-5 | $\mathcal{I}_{19}, \mathcal{I}_{20}, \mathcal{I}_{21}, \mathcal{I}_{22}, \mathcal{I}_{23}$ |
| PINFO | Continuous | 0-0.92 | \mathcal{I}_{24} |
| FORMAL | Ordinal | 1-5 | $\mathcal{I}_{25}, \mathcal{I}_{26}, \mathcal{I}_{27}, \mathcal{I}_{28}, \mathcal{I}_{29}$ |
| CENTRAL | Ordinal | 1-5 | $\mathcal{I}_{30}, \mathcal{I}_{31}, \mathcal{I}_{32}, \mathcal{I}_{33}, \mathcal{I}_{34}$ |
| COMPETE | Ordinal | 1-5 | $\mathcal{I}_{35}, \mathcal{I}_{36}, \mathcal{I}_{37}, \mathcal{I}_{38}, \mathcal{I}_{39}$ |
| PREDICT | Ordinal | 1-5 | $\mathcal{I}_{40}, \mathcal{I}_{41}, \mathcal{I}_{42}, \mathcal{I}_{43}, \mathcal{I}_{44}$ |

Table 3. Statistics of 50 pruned networks^a

| Before pruning | |
|---------------------------------------|------------------|
| Number of connections | 270 |
| Number of hidden nodes | 6 |
| Ave. accuracy on training set | 84.23 % (3.00 %) |
| Ave. accuracy on cross-validation set | 77.67 % (2.63 %) |
| Ave. accuracy on testing set | 78.56 % (2.29 %) |
| After pruning | |
| Average no. of connections | 10.30 (5.24) |
| Average no. of hidden nodes | 2.72 (1.16) |
| Ave. accuracy on training set | 75.81 % (0.61 %) |
| Ave. accuracy on cross-validation set | 78.72 % (2.06 %) |
| Ave. accuracy on testing set | 79.08 % (1.69 %) |

^aFigures in parentheses denote standard deviations.

Table 4. Accuracy of Rule 1

| SECTOR | Classification accuracy | Prediction accuracy |
|-------------------|-------------------------|---------------------|
| Overall | 393/510 (77.06 %) | 101/128 (78.91 %) |
| Transport | 40/54 (74.07 %) | 8/14 (57.14 %) |
| Wholesale | 138/192 (71.88 %) | 36/48 (75.00 %) |
| Retail | 87/118 (73.73 %) | 24/29 (82.76 %) |
| Banking & finance | 88/95 (92.63 %) | 21/24 (87.50 %) |
| Miscellaneous | 40/51 (78.43 %) | 12/13 (92.31 %) |

Table 5. Accuracy of Rule 2

| SECTOR | Classification accuracy | Prediction accuracy |
|-------------------|-------------------------|---------------------|
| Overall | 398/510 (78.04 %) | 106/128 (82.81 %) |
| Transport | 40/54 (74.07 %) | 9/14 (64.29 %) |
| Wholesale | 144/192 (75.00 %) | 40/48 (83.33 %) |
| Retail | 87/118 (73.73 %) | 24/29 (82.76 %) |
| Banking & finance | 88/95 (92.63 %) | 21/24 (87.50 %) |
| Miscellaneous | 39/51 (76.47 %) | 12/13 (92.31 %) |

Table 6. Accuracy rates obtained from discriminant analysis

| SECTOR | Classification accuracy | Prediction accuracy |
|-------------------|-------------------------|---------------------|
| Overall | 382/510 (74.90 %) | 98/128 (76.56 %) |
| Transport | 40/54 (74.07 %) | 11/14 (78.57 %) |
| Wholesale | 142/192 (73.96 %) | 34/48 (70.83 %) |
| Retail | 82/118 (69.49 %) | 22/29 (75.86 %) |
| Banking & finance | 82/95 (86.32 %) | 20/24 (83.33 %) |
| Miscellaneous | 36/51 (70.59 %) | 11/13 (84.62 %) |

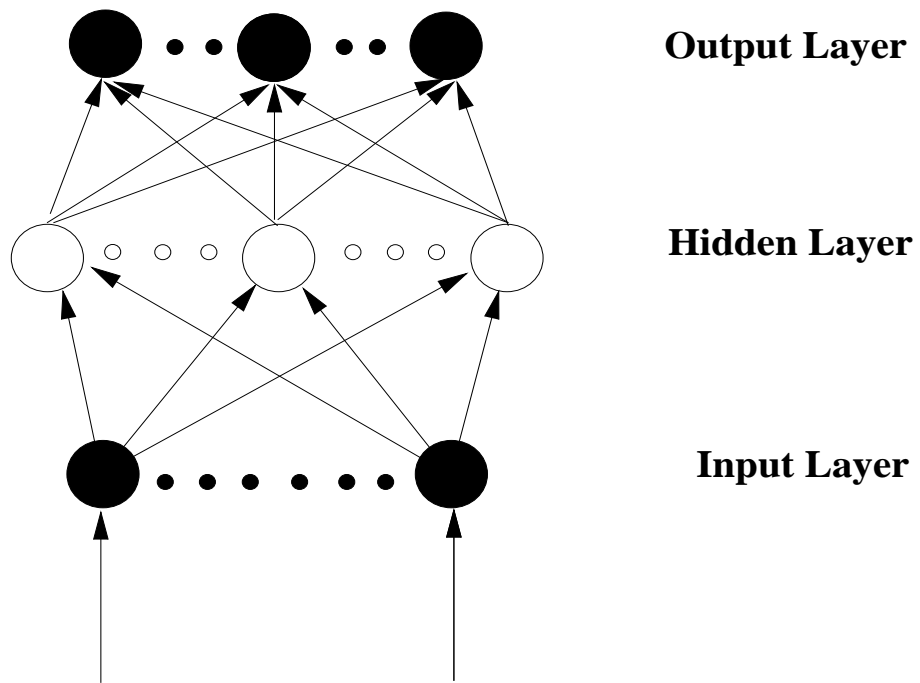


Fig. 1. A fully connected three-layer feedforward neural network

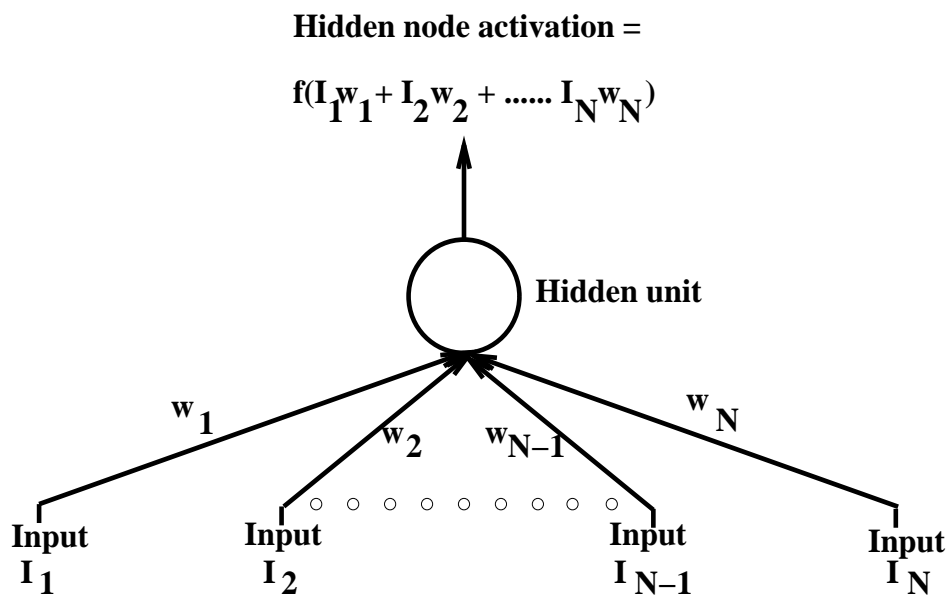


Fig. 2. Computing a hidden node activation value.

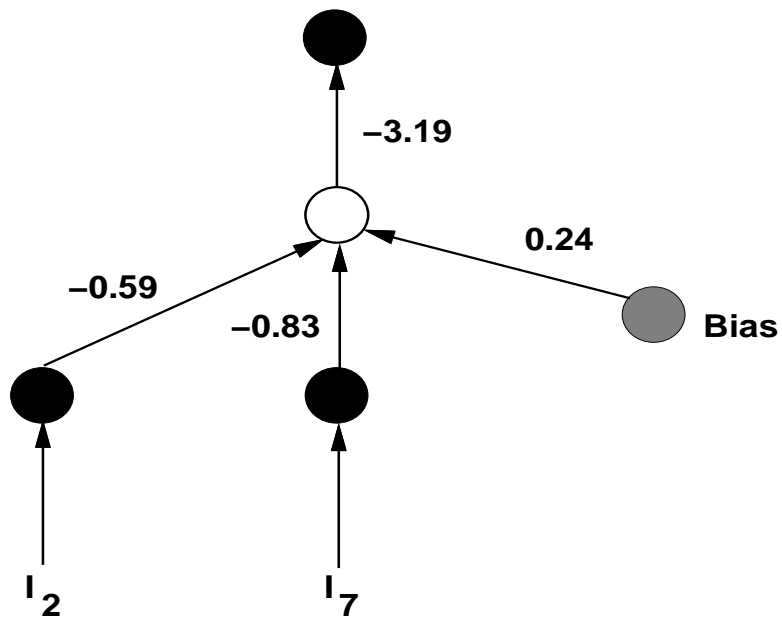


Fig. 3. Network in Example 1. The numbers shown are the connection weights.

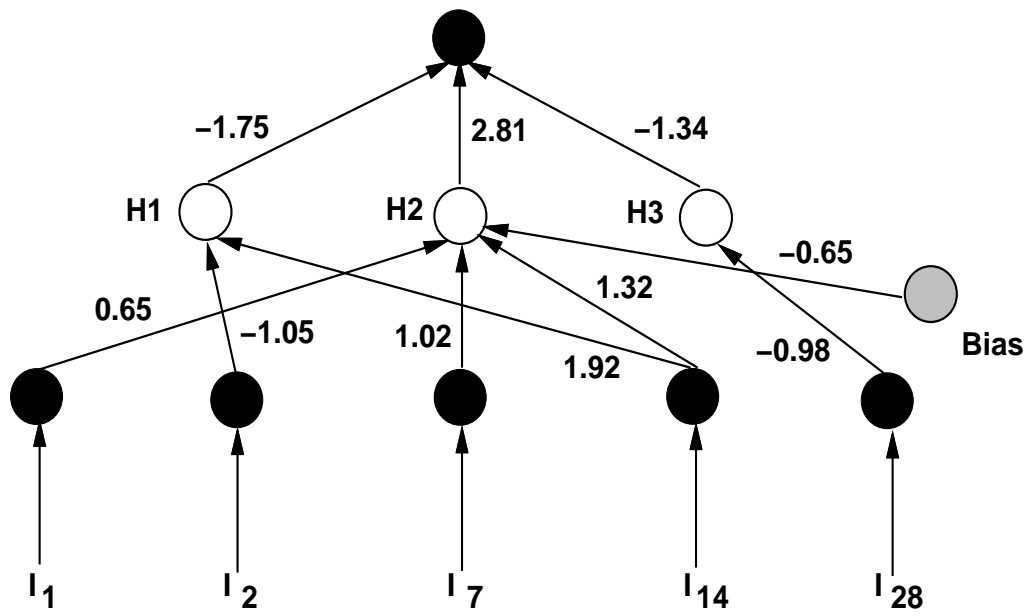


Fig. 4. Network in Example 2.