How Neural Networks Can Help Loan Officers to Make Better Informed Application Decisions

Meliha Handzic, Felix Tjandrawibawa and Julia Yeo
The University of New South Wales, Sydney, Australia

m.handzic@unsw.edu.au felixt@unsw.edu.au julia.yeo@unsw.edu.au

Abstract

The granting of loans by a financial institution (bank or home loan business) is one of the important decision problems that require delicate care. It can be performed using a variety of different processing algorithms and tools. Neural networks are considered one of the most promising approaches. In this study, optimal parameters and the comparative efficiency and accuracy of three models: Multi Layer Perceptron, Ensemble Averaging and Boosting by Filtering have been investigated in the light of credit loan application classification. The goal was to find the best tool among the three neural network models for this kind of decision context. The experimental results indicate that Committee Machine models were superior to a single Multi Layer Perceptron model, and that Boosting by Filtering outperformed Ensemble Averaging.

Keywords: knowledge discovery, classification, neural network, loan application

Introduction

The granting of loans by a financial institution (bank or home loan business) is one of the important decision problems that require delicate care. Loan applications can be categorized into good applications and bad applications. Good applications are the applications that are worthy of giving the loan. Bad applications are those ones that should be rejected due to the small probability of the applicants ever returning the loan. The institution usually employs loan officers to make credit decisions or recommendations for that institution. These officers are given some hard rules to guide them in evaluating the worthiness of loan applications. After some period of time, the officers also gain their own experiential knowledge or intuition (other than those guidelines given from their institution) in deciding whether an application is loan worthy or not.

Generally, there is widespread recognition that the capability of humans to judge the worthiness of a loan is rather poor (Glorfeld, 1996). Some of the reasons are: (i) There is a large gray area where the decision is up to the officers, and there are cases which are not immediately obvious for decision making; (ii) Humans are prone to bias, for instance the presence of a physical or emotional condition can affect the decision making process. Also personal acquaintances with the applicants might distort the judgmental capability; (iii) Business data warehouses store historical data from the previous applications. It is likely that there is knowledge hidden in this data, which may be useful for assisting the decision-making. Unfortunately, the task of discovering useful relationships or patterns from data is difficult for humans (Handzic, 2001). The reasons for such difficulties are the large volume of the data to be examined,

Material published as part of these proceedings, either on-line or in print, is copyrighted by Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the publisher at Publisher@InformingScience.org

Paper Accepted as a Regular Paper

and the nature of the relationships themselves that are not obvious.

Given the fact that humans are not good at evaluating loan applications, a knowledge discovery tool thus is needed to assist the decision maker to make decisions regarding loan applications. Knowledge discovery provides a variety of useful tools for discovering the non-obvious relationships in historical data, while ensuring those relationships discovered will generalize to the new/future data (Bigus, 1996; Marakas, 1999). This knowledge in the end can be used by the loan officers to assist them in rejecting or accepting applications. Past studies show that even the application of a simplistic linear discriminant technique in place of human judgment yields a significant, although still unsatisfactory increase in performance (Glorfeld, 1996).

Treating the nature of the loan application evaluation as a classification (Smith, 1999) and forecasting problem (Thomas, 1998), it is argued here that neural networks may be suitable as knowledge discovery tools for the task. Therefore, the main objectives of this study are: (i) To develop a robust knowledge discovery tool using neural network models that is both reliable and easy to build, and (ii) To compare the performance of the basic neural network model called Multi Layer Perceptron (MLP) with Committee Machine models (namely Ensemble Averaging and Boosting by Filtering) in scoring credit loan applications.

Literature Review

Neural Networks

The initial work on neural networks was motivated by the study on human brain and the idea of neurons as its building blocks. Artificial intelligence researchers introduced a computing neuron model (depicted in Figure 1.) to simulate the way neurons work in human brain. This model provided the basis for many later neural networks developments.

There are literally hundreds of neural network models available to use (FAQ, 2002). The combination of topology, learning paradigm and learning rules/algorithms define a particular neural network model. The topology can be in the form of feed forward, limited recurrent and fully recurrent networks. The learning paradigm can be classified as one of these: supervised, unsupervised and reinforcement. There are different learning algorithms for neural networks, for example: error correction learning, Hebbian learning, competitive learning and Boltzman

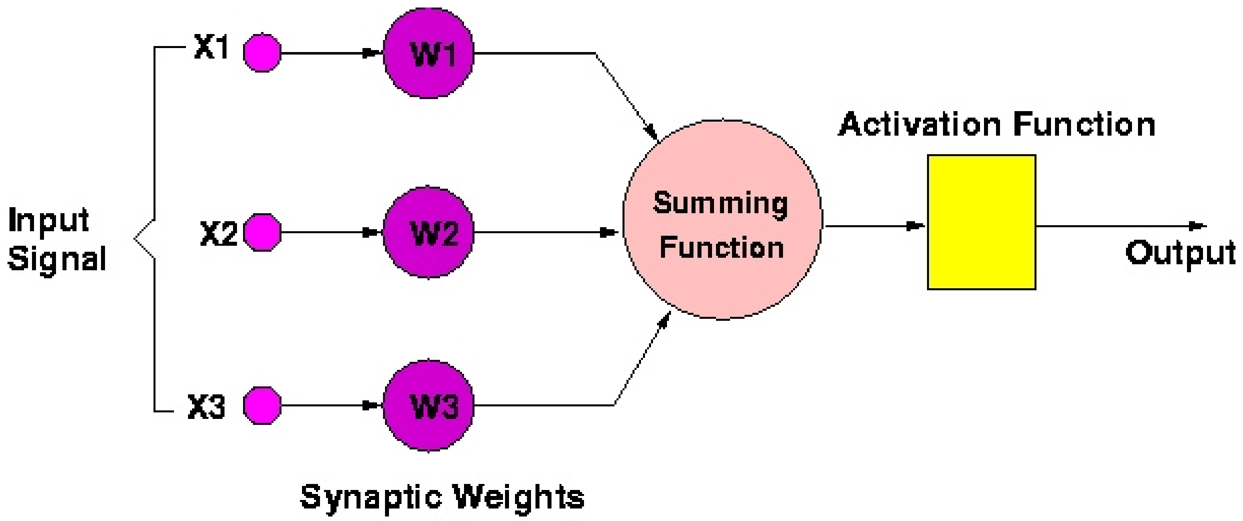

Figure 1 Model of Computing Neuron

learning. Back propagation method and Kohonen feature maps are most popular in knowledge discovery (Bigus, 1996).

The most important feature of neural networks is their ability to learn. Just like human brain, neural networks can learn by example and dynamically modify themselves to fit the data presented. Furthermore, neural models are also able to learn from very noisy, distorted, or incomplete sample data, which render other methods useless (Glorfeld, 1996). Because of this learning ability, neural networks are then seen as appropriate tools for knowledge discovery. Learning paradigms in neural networks are divided into 2 categories, supervised and unsupervised. With supervised learning, the model is provided with feedback so that it can learn from its mistakes and try not to repeat the same mistakes in the future. In the unsupervised learning, the neural network model is not given any prior training or feedback. Instead it is given real data from the beginning and will learn along the way.

Beside its ability to learn patterns, neural networks also have other useful properties and capabilities including: (i) Nonlinearity. A neural network, made up of an interconnection of non-linear neurons in it is not linear. Nonlinearity is an important property particularly if the input pattern is inherently nonlinear; (ii) Adaptivity. Neural networks have the ability to adapt their synaptic weights to changes in the surrounding environment. This is particularly useful when a neural network model is deployed in a non-stationary environment (an environment that changes with the time, for example a stock market price); and (iii) Generalization. This is perhaps the second most important feature besides learning. Generalization refers to the neural network producing appropriate outputs for inputs that were not encountered during training.

Despite the tremendous benefits offered by knowledge discovery for businesses, neural networks are not free from criticism. Most neural networks are of the "black box" kind. This means that the tool can only produce conclusions without explanations and justification of the reasons behind such conclusions. This makes accountability and reliability issues difficult to address. That is why one of the main interests in knowledge discovery research is to find ways to justify and to explain the knowledge discovery result. Other limitations concern the high computational requirements of neural networks, usually in the form of computer power and training time, and the scarcity of experts in the field, which makes some businesses avoid their use (Marakas, 1999).

Choosing which neural network model to use is not trivial. According to Bigus (1996), there are several criteria that can be followed in order to choose a suitable model. These criteria include data type and quantity, training requirements and functional requirements. Training requirements are mostly concerned with things such as: the tolerable training time and hardware demands for doing such training. Functional requirements are related to the learning function that is expected.

Credit Loan Evaluation

Most of the research on the application of neural networks to credit loan evaluation problem has focused on Multi Layer Perceptron (MLP). The empirical results are mixed. While some studies reviewed here indicate that a neural network approach is better than other techniques (Nittis et al. 1998, Malhorta et al. 2001), other studies suggest otherwise (Galindo et al. 1997, Desai et al. 1997, Yobas et al. 1997). This makes it hard to draw any general conclusions that would guide practice. The tendency for different experts to argue that they have the best model, whatever model that is, makes this problem even harder (Thomas 1998).

On a positive side, some studies reinforce Glorfeld's (1996) claim that human performance can be outperformed by simple statistical methods. Nittis et al. (1998) used a neural network as a second level filter by supplying it with data that already had been approved by loan officers. Their experiments show that the neural network developed was reliable enough to be the first level filter (replacing loan officers) with only a few cases needing human intervention. Furthermore, Malhorta et al (2001) discovered a marginal advantage using neural networks over discriminant analysis in classifying consumer loans, but the overwhelmingly convincing consistency of neural networks. Also, neural networks outperformed discriminant analysis in recognizing "bad" applications.

In contrast, several studies reported somewhat inferior performance of neural networks compared to other models or found no significant advantage. Galindo et al. (1997) in their empirical study examined 4 different techniques: classification and regression trees (CART), neural network models, k-nearest neighbor and Probi. Neural network models came second after CART in their experimental results. However, the difference in performance between them was small. Desai et al. (1997) concluded that the neural network (feedforward model) involved did not significantly outperform the conventional techniques because the most appropriate variants of the techniques were not used. Yobas et al. (1997) came to a similar conclusion with respect to credit card applications.

Although this review shows that neural networks may not always be the best possible tool for loan application evaluation, it also reveals that it has never been more than marginally outperformed by other methods. The reasons why neural networks do not always come out as a winner is these comparison studies are: (i) The most appropriate variants of neural network are not used (Desai et a. 1997). There is still a wide variety of neural network models that have not been tested against the loan application problem; (ii) The studies reviewed above each used different data. For example, Galindo et al. (1997) used 24 attributes and 4000 instances, while Malhorta et al. (2001) used data with 6 attributes and 700 instances. The characteristics of the data, for example number of attributes and the distribution of the data, are all important as they affect the performance of the developing neural network.

Research Objectives

Based on the above review and discussion, this study was set to investigate, on the same collection of real loan data, the comparative performance of two different Committee Machine models and the basic Multi Layer Perceptron. Ensemble Averaging was suggested to be successful in reducing bias and variance of a single expert model (Perrone, 1993; Volker 2001). Similarly, Boosting by Filtering was proposed to boost learners' performance by enabling individual experts to learn from others' mistakes (Shapire 1990, Haykin 1999). While it was expected that both committee machines would perform reasonably well in loan application evaluation, no prior empirical research was done to establish this.

The current study was carried out in two stages. First, a common MLP model was built and tested for predicting worthiness of loan applications. Then, based on the MLP model built, the study concentrated on developing and testing two static committee machine models, Ensemble Averaging and Boosting by Filtering, to see how much improvement they would provide over MLP. These two models were chosen among other static committee machine models because they were both easy to implement and had the theoretical potential to improve performance over MLP.

Research Method

KDD Tool Design Overview

The knowledge discovery tool built in this research project consists of 5 files, each representing a class on its own. The relation of these classes can be seen on the simplified Unified Modeling Language (UML) diagram shown in Figure 2. Briefly, MLP implementation was built upon two main modules, BPNN and mlp. BPNN provides backpropagation algorithm that is then used by mlp. NeuralGUI is an extension of MLP implementation, providing a user with graphical interaction with the program. Committee and boosting modules represent the committee machines implementation. They use MLPs as experts.

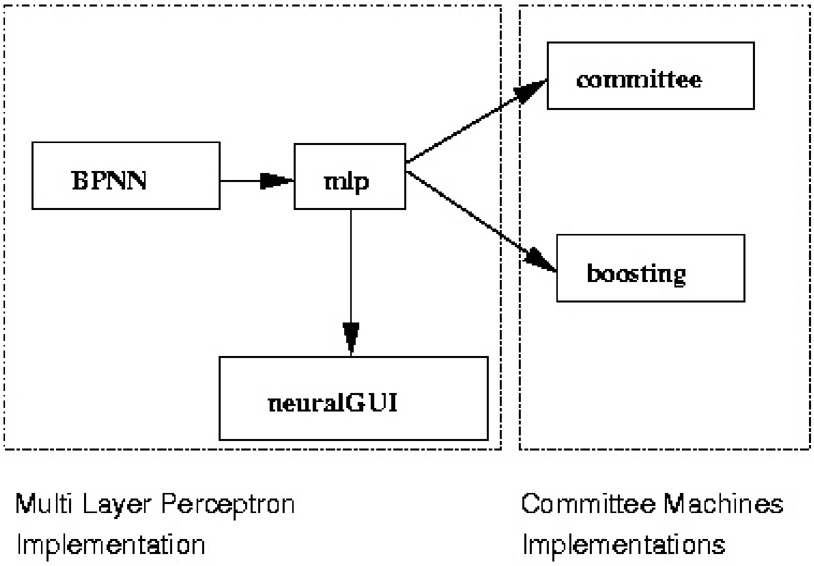
BPNN
mlp
neuralGUI
committee
boosting
Multi Layer Perceptron Implementation
Committee Machines Implementations

Figure 2 Knowledge Discovery Tool Structure

In designing the tool, the following considerations had to be taken into account. An MLP with 1 output neuron can produce 2 different results (binary decision: 0 or 1). Combined with two expected results, bad application and good application, we can have four different combinations of possible prediction result. Thus, result of classification can be divided into 4 possible outcomes: (i) Negative correctly classified result, the network is rejecting a bad application; (ii) Negative incorrectly classified result, the network is accepting a bad application; (iii) Positive correctly classified result, the network is accepting a good application; and (iv) Positive incorrectly classified result, the network is rejecting a good application.

From these four, we argue that the most important one is the negative misclassified application (highlighted in Table 1). Negative data (that is non-loan worthy application) that is misclassified as loan worthy has a more damaging impact than the other types of misclassification. If negative data is misclassified, that means, the bank is losing money due to granting loans to wrong applicants. Thus, our experiments were focused on this set of misclassifications and trying to improve the performance of the neural net model in this particular area. However, it is important to recognize that in achieving low error on classifying negative data, there is a cost to pay. There is a trade off between pursuing best performance on negative misclassified data and pursuing the best performance on all data. That is, in trying to improve the network performance in identifying the negative data, the performance on positive data is going to be worsened.

Multi layer perceptron implementation

As mentioned earlier, the first step in this investigative study involved building and testing of the performance of

Table 1: Result of Loan Classification

	Results	Application Outcome
Output Neuron Generator	Positive	Good application
		Bad application
	Negative	Good Application
		Bad Application

the most widely used neural network model called Multi Layer Perceptron (MLP) in dealing with credit loan applications. The result of this MLP implementation served as the base for later implementations of combined neural networks.

The design of the MLP module was an adaptation from Tom Mitchell's MLP suite (Mitchell, 1997). BPNN class provided the functionality of backpropagation algorithm, while mlp class provided an interface to integrate these functionalities with the functionalities found in Data module. As the backpropagation learning algorithm requires randomization of initial weights, it was necessary to provide some sort of randomization methods in the program. This was done in such a way to allow the seed to be determined by the user. It is important to notice that, the ability to set the seed and thus control the initial weights for the network is necessary for reproducing the experiment. Without having the mechanism of reproducing a set of random numbers every time, it would be difficult to compare changes between experiments.

As explained before, the study was conducted with the goal to find a balance in performance trade off. To achieve the desired performance balance, two threshold variables were included in the program, namely a maximum tolerable error percentage on all data and on negative data. These two thresholds provided an early stopping point for training. Finally, to assist with the experiment, a Graphical User Interface (GUI) was also provided. Most commercial knowledge discovery software has GUIs to assist the user of the system. As this project was trying to build a knowledge discovery tool that was not only robust but also user-friendly, a simple GUI was thus implemented. The GUI makes it easier for the user to test the constructed neural network model, and the parameters of MLP can be easily set using the GUI.

Ensemble averaging implementation

Ensemble Averaging was implemented as a combination of several experts, where the individual experts combined were MLPs developed in the earlier stage of the study. Appropriate program modules were built to store and read user specifications, to add an arbitrary number of experts, to train experts and to combine their assessments. For evaluating a loan application, each expert produces its own assessment on the particular application, and then the combiner takes the average from these assessments to make the final assessment on the loan application.

Boosting by filtering implementation

Boosting by Filtering was implemented using a simplified version of the boosting algorithm described by Haykin (1999). The simplified algorithm differs from the original in choosing the training data for the second expert. In the original algorithm, the first expert keeps on discarding training instances until it finds the instance suitable for training the second expert and then the process is repeated until sufficient number of training instances are reached. In the simplified version, the first expert only needs one pass to gather the necessary instances for training the second expert, and it is done by adding instances from the same set until the set has no more instances.

Experimental Data

In relation to loan application evaluation, the information that is regarded as important (Thomas, 1998) includes: (i) The character of the applicant, credit history of the applicant and his/her family is certainly helpful; (ii) The capital, this identifies how much money is being asked by the applicant; (iii) The collateral, this refers to what things that the applicant willing to put up from his/her own resources; (iv) The capacity, the applicant's repaying ability is one of the most crucial factors, this information can be in the form of how much income does he/she earn, how many jobs does he/she have or how long has he/she been working in their job; and (v) The condition of current economic market.

The actual data used for this study was taken from the UCI Repository; this data is publicly available and can be accessed from the URL: ftp://ftp.ics.uci.edu/pub/machine-learning-databases/credit-screening. The data set was compiled by Chiharo Sano and was generated in accordance to input from a Japanese company that grants credit. All attribute names and values were changed to meaningless symbols to protect confidentiality within the data. However, it was expected that most of the important information (as described earlier) was included in the data set. The last column/attribute was the outcome of the application. If it was "+", that meant the application was a good application. On the other hand, the "-" sign indicated that the application was a bad one. The data set had a total of 690 instances. Of these, 307 instances (44.5%) were good applications, where 383 (55.5%) instances were bad applications. In addition, 37 instances, that is, 5% had one or more missing attribute values.

Data pre-processing

Data pre-processing is the first step of the knowledge discovery process. In this step, data in the data warehouse was preprocessed and then fed to the knowledge discovery tool. Since neural network models only accept discrete or continuous numerical values, categorical data is needed to translate into numerical values for it to be utilized by the neural network models. This was done using the 1-out-of-N encoding technique (Smith, 1999). The numerical data needs to be normalized in order for it to be used by the neural network. Normalization in this case meant dividing an input vector with a norm of the vector. It is a common practice to normalize the input of MLP model to the range of 0 to 1, but there are cases when scaling the inputs to the range of -1 to 1 is a considerably better option (FAQ, 2002). The normalization used in this project is called rescaling, and was done by finding the maximum value of particular numerical attributes and then dividing each of the instances of the attribute by this maximum.

As mentioned earlier, the data set contained 37 instances with one or more missing attribute values. This is sometimes referred to as the noise in the data. Since neural networks have been known for their capacity to handle noisy data, these instances were included as valid cases in the current investigation. The historical data for loan applications after being preprocessed was stored in a file. A data structure was needed to represent the information stored in this file during the running of the program. The data structure was described in the Data module. The Data class encapsulated the application data in the form of lists of values in a Data object. The data object also stored other information such as the number of negative applications, the number of instances etc.

To prevent the MLP model becoming biased towards the data set used for training, a data partitioning technique was used. The commonly used technique is called cross validation. It is basically an approach where the input data is divided into two sets; one set is used for training the MLP model as can be referred as the training set. The rest is for validating the performance of the model; this is called the test set. Training set is used for the model to converge to an empirical optimal, and the other set is used to decide when to terminate the convergence process (Perrone, 1993). This study applied 50:50 divisions (346 for training set and 344 for data set). Both training and test sets contained half negative data and half positive data. The training set having at least half of all its data negative allowed the network to learn to classify the negative data properly (Smith, 1999). The special program module provided a function to create two sets that had equal percentage of negative data.

As is often the case with many boosting algorithms, a method to apply random sampling was required. The samples were generated from the list from which the Data object was created. The sample was taken with replacement, thus in the resulting set, duplicates were possible. Finally, an appropriate method was provided to produce statistical information on a data column. This information consisted of the sum of all the column values, average values, maximum and minimum values and was used during the preprocessing of the data.

Experimental Results

Optimal MLP Parameters

Finding the best parameters for MLP model is a crucial issue. The optimal MLP would have a combination of parameters that minimize the classification error. The goal of the MLP experiments therefore was to find these combination parameters that were best for evaluating loan applications. In determining optimal MLP parameters, average values from a series of repeated experiments were used. Taking the average results is an approach applied to lessen the instability inherited by the MLP model.

Number of hidden neurons

The aim of the first series of experiments was to find the number of hidden neurons needed for the network to be able to evaluate loan applications with the highest accuracy possible given a fixed number of training epochs. The experiments started by using one hidden neuron, then the number of hidden neurons was increased until the performance improvement was no longer observed. For each experiment, six runs of the network were used; each run had different weight initialization. The result of a particular experiment was the average from the runs. A run was finished when the training epochs reached 500.

An MLP with 15 hidden neurons had the best performance compared to other numbers of hidden neurons. The study found that the network reached its peak performance when 15 hidden neurons were used and then the network suffered a decrease in performance when more hidden neurons were added. This result confirms the theory that having too many or too few neurons in a hidden layer can have a negative effect on the network performance.

Weight initialization

The next series of experiments was conducted to test the extent of performance variations due to weight initialization. In theory, the variations of initial weights will result in variations in network performance. Ten experiments were conducted, each using MLP with different weight initializations. These experiments were measuring the number of training periods needed to reach a desired error percentage.

From the 10 experiments conducted, one of the networks was unable to meet the thresholds after 1500 epochs, while the other 9 were able to do so with an average of 716.4 epochs. From this finding, the significance of weight initialization is apparent. A network with bad weight initialization results in slower converge (more training epochs are needed); in fact there is no guarantee that this network will converge at all to the performance thresholds. Despite this, the chance of a network in getting the “wrong” weights initialization is quite small (1 out of 10 in this experiment). Our results confirm Smith’s (1999) finding that the effect of weight initialization for most applications is not significant.

Momentum

The aim of the momentum experiments was to find momentum that effectively helps the network to avoid local minima and speed up the convergence. For each experiment there were five runs, each run with different weight initialization. With the error threshold set on, a network would terminate training when the error percentage falls below the threshold, or when after 1000 epochs the network was still unable to reach the desired performance.

The results indicate that having a momentum value in the range between 0.6 and 0.7 contributed to speeding up the network's pace of learning. When momentum was set to 0.8, few runs were unable to meet the thresholds after 1000 epochs. For the rest of experiments 0.6 was used as an optimal value. Setting the momentum to above 0.7 makes the network too volatile causing it to fail to descend to better minima.

Parameters	Value
Number of Data in Training Set	346
Number of Data in Test Set	344
Number of Hidden Neurons	15
Seed	0
Learning Rate	0.2
Momentum	0.6
Training epochs	1000

Table 2 Optimal MLP Configuration

Learning rate

Several experiments were carried out to find a learning rate that effectively controls the extent of weight modification during training epochs. For each experiment, there were six runs of the network. Then the average epochs needed to reach the desired threshold value were calculated. If after 1000 epochs, the network is still unable to reach the desired performance, the training would be stopped.

The results shows a somewhat non-linear behavior of the impact of learning rate on performance, hence it is difficult to find a general trend. In later experiments (after experimenting with a learning rate of 0.5), some of the experiments did not converge within the 1000 epochs. Interestingly, some earlier research using a learning rate greater than 0.5 produced networks which required 150 training epochs. This suggests that a learning rate greater than 0.5 causes the network to have volatile performance, hence for loan application problem it is desirable to use a learning rate lower than 0.5. For the rest of this study the learning rate was set at 0.2. This value was chosen as it is close to what commercial neural network packages use (Smith, 1999). Given the above justifications for choosing the parameters for MLP model, an optimal configuration is presented in Table 2.

Credit Scoring Performance

After determining the optimal parameter combination for the MLP model, the next step was to determine its credit scoring performance and compare it with that of the two committee machines. The credit scoring performance of all three models was measured against the test data and was evaluated in terms of their accuracy and speed.

Two measures of accuracy used were percentage error on negative data and percentage error on all data. These metrics were used to assess the ability of the network to reduce error. Percentage error on negative data measures misclassified negative data against the number of negative data in the set. Percentage error on all data measures misclassified data (both positive and negative) against the number of all data in the set.

The two measures of speed were number of epochs and training time. These metrics were used to assess how fast the network could learn, and also how much training was needed for the network to perform according to the training requirements. The results are as follows.

MLP results

The combination of parameters presented in Table 1, provided us with an MLP model that is customize d for predicting bad loans, and at the same time performance in predicting or classifying other classes of loan applications is not overly sacrificed.. The test results indicate that the average percentage errors achieved by MLP were 1.81% on negative data and 2.38% on all data. This was achieved in less than 1000 epochs (600) and in a short training time (13 sec). These results are better than Smith (1997) whose experiment produced MLP that was able to classify bad applications correctly 46.7% of the time and good applications 90.9% of the time.

Ensemble averaging results

A static committee machine model called Ensemble Averaging was the subject of the following experiment. It is argued that ensemble averaging brings stability in performance. Different weights can lead to better or worse performance and ensemble averaging compromises the two extremes by lessening the e ffect of choosing "wrong" weight combinations. It has been proven to be able to enhance MLP performance in other applications, like medical reports. The purpose of the current study was to find out the degree of improvement that Ensemble Averaging could provide over MLP in the context of loan application evaluation. The committee machine was built upon the ten experts from MLP experiments. The combiner program used simple voting for combining the results from these experts. As a committee, the model achieved 1.73% error on negative data, and 2.06% on all data. However, it came at the cost of increased training time (245 sec).

Boosting by filtering results

To test the performance of the Boosting by Filtering committee machine in classifying loan applications, three experts were used. Each of the experts was MLP with optimum configuration based on the result from MLP experiments. Each expert was trained with 500 epochs. The first expert on average produced 278 training instances for the second expert. The first and second expert together on average produced 38 training instances for the third expert. This shows that there were roughly 38 cases that were hard to classify and the third expert concentrated on these cases. The average performance of Boosting by Filtering indicated small error scores of 1.32% on negative data and 1.65% on all data within a reasonable training time (32 sec).

Comparative performance analysis

The overall results of the study are presented in Table 3 and clearly suggest that MLP performance in classifying loan applications can be further improved by committee machines models. In particular, Ensemble Averaging

Neural Network Model	Number of Epochs	Percentage Error on Negative Data	Percentage Error on All Data	Average Training Time
Multi Layer Perceptron	600	1.81 %	2.38 %	13 sec
Ensemble Averaging (10 MLPs)	6800	1.73 %	2.06 %	245 sec
Boosting by Filtering (3 MLPs)	1500	1.32 %	1.65 %	32 sec

Table 3 Comparative Model Performance

has been shown to be able to reduce the percentage error due to the bias-variance problem inherited by MLP model. However, on average, the improvement ensemble averaging brings on negative data classification is not great. This is evidenced by marginal 0.09% improvement on negative data. The performance on all data was more convincing with 0.32% improvement. While ensemble averaging was able to produce lower percentage errors, it did so at the cost of training time, as evident in the training time for this model compared to other models.

Furthermore, the results indicate that Boosting by filtering outperformed other models in this study. It was able to improve the performance of MLP model by 0.49% on negative data and 0.73% on all data. The boosting by filtering committee machine was the best performer in this experiment, in that it was able to produce the least percentage error. This was done at a comparatively low training time cost.

Overall, the small percentage error produced by neural network models in this experiment (1.32% - 1.81% average percentage error on negative data and 1.65%-2.38% average percentage error on all data) confirms that neural network models are well suited for loan application evaluation. The boosting by filtering committee machine shows that training different experts on hard to classify applications brings a significant performance improvement. Boosting by filtering also shows that these performance improvements can be achieved at a low cost (less training time and computational cost).

Conclusion

This research project was concerned with the development and experimentation of a knowledge discovery tool using neural network models in relation to loan application evaluation. Given the short time available to do the project, a knowledge discovery tool has been built, the tool consisting of 3 neural network models: a Multi Layer Perceptron, an Ensemble Averaging committee machine and a Boosting by Filtering committee machine. The tool was tailored for loan applications evaluation. The experimental results confirm that committee machines are able to perform in superior manner compared to MLP, with Boosting by Filtering outperforming the Ensemble Averaging model. With their high accuracy in classifying loan applications, all neural network models implemented in this project, can certainly be helpful for the decision making process. However, more trust is necessary before neural networks can gain broader acceptance in business world.

Despite interesting findings, the study faced some limitations. The main difficulty was in determining the best possible set of model parameters. For instance, the model requires foundational knowledge of consumer behavior to be factored in. Neural network models have a very high degree of freedom. What this means is that there is a wide range of different combinations of parameters that can affect the performance of a particular network. This causes difficulties because there are still no really fixed and justified ways to set the values of parameters, so it is expected that the user, the company or the decision making team will need a period time to find the most suitable or near perfect parameters of any given neural network model. Future studies could include longitudinal studies of 'successful' loan applications so as to analyze the factors of repayment success. Also, it would be interesting to conduct a comparative study of positive data misclassified as bad application versus negative data misclassified as good application.

A newly developed genetic algorithm has been proposed to be a solution for parameter selection problem. Integrating such elements into the knowledge discovery tool should enhance the performance of the tool (Marakas, 1999). Furthermore, a genetic algorithm may improve the efficiency of the knowledge discovery tool as it should provide an automated procedure to find better parameter selections. In addition, providing a visualization component to obtain a more sophisticated Graphical User Interface (GUI) would be a useful thing to pursue, as it would provide the decision maker with a greater justification of network performance. The explanatory capability of neural networks is still a weak point. A knowledge discovery tool using a neural network that is also self-explanatory would certainly be of major interest to the business world.

References

Bigus, J.P. (1996). *Data mining with neural networks: Solving business problems from application development to decision support*. McGraw Hill, USA.

Desai, V.S., Convay, D.G., Crook, J.N. & Overstree, G.A. (1997). Credit scoring models in the credit union improvement using neural networks and genetic algorithms. *IMA J Mathematics Applied in Business and Industry, 8*, 323-346.

FAQ. (2002). Retrieved June 5, 2002, from http://ftp.sas.com/pub/neural/FAQ.html

Galindo, J. & Tamayo, P. (1997). Credit risk assessment using statistical and machine learning methods as an ingredient for financial intermediaries risk modeling. Retrieved June 5, 2002, from http://www.defaultrisk.com/pp

Glorfeld, L.W. & Hardgrave, B.C. (1996). An improved method for developing neural networks: The case of evaluating commercial loan creditworthiness. *Computer Operation Research, 23* (10), 933-944.

Handzic, M. & Aurum, A. (2001). Knowledge discovery: Some empirical evidence and directions for future research. *Proceedings of the 5th. International Conference on Wirtschafts Informatics (WI'2001)*, 19-21 September, Augsburg, Ge rmany.

Malhorta, R. & Malhorta, D.K. (2001). Evaluating consumer loans using neural networks. Retrieved June 5, 2002, from http://www.efmaefm.org/MalhortaDavinder/MalhortaDavinder1.pdf

Marakas, G.M. (1999). *Decision support systems in the twenty-first century*. Prentice Hall, New Jersey, USA.

Mitchell, T. (1997). *Machine learning*. McGraw Hill, USA.

Nittis, F.D., Teachioli, G. & Zorat, A. (1998). Consumer loan classification using artificial neural networks. Retrieved June 6, 2002, from www.neuricam.com/neuricam/docs/papers/EIS98.pdf

Smith, K.A. (1999). *Introduction to neural networks and data mining for business applications*. Eruditions Publishing, Australia.

Thomas, L.C. (1998). A survey of credit and behavioural scoring: Forecsting financial risk of lending to customers. Retrieved June 5, 2002, from www.bus.ed.ac.uk/working_papers/full_text/crc9902.pdf

Yobas, M.B., Crook, J.N. & Ross, P. (1997). Credit scoring using neural and evolutionary techniques. Working Paper 97/2, Credit Research Centre, University of Edinburgh. Retrieved June 5, 2002, from http://www.dai.ed.ac.uk/papers/documents/mt9687.html

Haykin, S. (1999). *Neural networks: A comprehensive foundation*. Prentice-Hall, New Jersey, USA.

Shapire, R.E. (1990). The strength of weak le arnability. *Machine Learning, 5*, 197-227.

Volker, T. (2001). Committee machines. *Handbook for Neural Network Processing*. CRC Press.

Perrone, M.P. (1993). *Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization*. PhD Thesis, Brown University, Rhode Island. Retrieved June 5, 2002, from http://citeseer.nj.nec.com/perrone93improving.html

Biography

Dr. Meliha Handzic is a senior academic at the School of Information Systems, Technology and Management, University of New South Wales. She is the founder and the group leader of Knowledge Management Research Group (kmRg). Her main research interest is in the area of Knowledge Management and Decision Support. Meliha has over 50 research papers published on these topics.

Felix Tjandrawibawa is an undergraduate student in Computer Science and Engineering at the University of New South Wales. His particular interest lies in research, development and application of neural networks in decision making tasks.

Julia Yeo is an associate lecturer and a PhD student at the School of Information Systems, Technology and Management, University of New South Wales. Her research interests include knowledge management, information privacy as well as IT education.