Expert Systems with Applications 37 (2010) 6233–6239

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes

Adnan Khashman *

Intelligent Systems Research Group (ISRG), Near East University, Lefkosa, Mersin 10, Turkey

ARTICLE INFO

Keywords:
Credit risk evaluation
Artificial neural networks (ANN)
Back-propagation algorithm
Neural learning schemes
Finance and banking

ABSTRACT

This paper describes a credit risk evaluation system that uses supervised neural network models based on the back propagation learning algorithm. We train and implement three neural networks to decide whether to approve or reject a credit application. Credit scoring and evaluation is one of the key analytical techniques in credit risk evaluation which has been an active research area in financial risk management. The neural networks are trained using real world credit application cases from the German credit approval datasets which has 1000 cases; each case with 24 numerical attributes; based on which an application is accepted or rejected. Nine learning schemes with different training-to-validation data ratios have been investigated, and a comparison between their implementation results has been provided. Experimental results will suggest which neural network model, and under which learning scheme, can the proposed credit risk evaluation system deliver optimum performance; where it may be used efficiently, and quickly in automatic processing of credit applications.

1. Introduction

Credit risk analysis is an important topic in financial risk management, and has been the major focus of financial and banking industry. Credit scoring is a method of predicting potential risk corresponding to a credit portfolio. Models based on this method can be used by financial institutions to evaluate portfolios in terms of risk. Data mining methods, especially pattern classification, using real-world historical data, is of paramount importance in building such predictive models (Yu, Wang, & Lai, 2008).

Due to financial crises and regulatory concerns of the Basel Committee on Banking Supervision (2000, 2005), a regulatory requirement was made for the banks to use sophisticated credit scoring models for enhancing the efficiency of capital allocation. The Basel Committee, comprised of central bank and banking business representatives from various countries, formulated broad supervisory standards and guidelines for banks to implement. Due to changes in the banking business, risk management practices, supervisor approaches, and financial markets, the committee published a revised framework as the new capital adequacy framework, also known as Basel II (Basel Committee on Banking Supervision, 2005). The commencement of the Basel II requirement, popularization of consumer loans and the intense competition in financial market has increased the awareness of the critical delinquency issue for financial institutions in granting loans to potential applicants (Li, Shiue, & Huang, 2006).

Credit scoring tasks can be divided into two distinct types (Laha, 2007; Li et al., 2006; Vellido, Lisboa, & Vaughan, 1999). The first type is application scoring, where the task is to classify credit applicants into "good" and "bad" risk groups. The data used for modeling generally consists of financial information and demographic information about the loan applicant. In contrast, the second type of tasks deals with existing customers and along with other information, payment history information is also used here. This is distinguished from the first type because this takes into account the customer's payment pattern on the loan and the task is called behavioral scoring. In this paper, we shall focus on application scoring.

In credit scoring; a scorecard model lists a number of questions (called characteristics) for loan applicants who provide their answers based on a set of possible answers (called attributes). As a credit scoring method, neural network models are quite flexible as they allow the characteristics to be interacted in a variety of ways. They consist of a group or groups of connected characteristics. A single characteristic can be connected to many other characteristics, which make up the whole complicated network structure. They outweigh decision trees and scorecards because they do not assume uncorrelated relations between characteristics. They also do not suffer from structural instability in the same way as decision trees because they may not rely on a single first question for constructing the whole network. However, the development of the network relies heavily on the qualitative data that are solicited

* Tel.: +90 392 223 6464; fax: +90 392 223 6622.
E-mail addresses: amk@neu.edu.tr, khashman@ieee.org
URL: http://isrg.neu.edu.tr

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.02.101

to specify the interactions among all characteristics (Cheng, Chiang, & Tang, 2007).

The use of neural networks in business applications has been previously investigated by several works (Ahn, Cho, & Kim, 2000; Baesens, Gestel, Stepanova, Van den Poel, & Vanthienen, 2005; Baesens, Sentiono, Mues, & Vanthienen, 2003a; Becerra-Fernandez, Zanakis, & Walczak, 2002; Hsieh, 2005; Huang, Chen, Hsu, Chen, & Wu, 2004; Huang, Nakamori, & Wang, 2005; Lee & Chen, 2005; Lee, Chiu, & Lu, 2002; Malhotra & Malhotra, 2003; Min & Lee, 2005; Smith, 1999; Vellido et al., 1999; West, Dellana, & Qian, 2005). The general outcome of such works is that in the credit industry, neural networks have been considered to be accurate tool for credit analysis among others (Min & Lee, 2008).

Recently, the work in Lim and Sohn (2007) proposed a neural network-based behavioral scoring model which dynamically accommodates the changes of borrowers' characteristics after the loans are made. This work suggested that the proposed model can replace the currently used static model to minimize the loss due to bad creditors. In (Martens, Baesens, Van Gestel, & & Vanthienen, 2007), an overview of rule extraction techniques for support vector machines when applied to medical diagnosis and credit scoring was presented. This work proposed also two rule extraction techniques taken from the artificial neural networks domain. In (Huang, Chen, & Wang, 2007), hybrid SVM-based credit scoring models were proposed to evaluate an applicant's credit score from the applicant's input features. This work used the Australian and German datasets in its implementation.

More recently, the work in Abdou, Pointon, and Elmasry (2008) investigated the ability of neural networks, such as probabilistic neural nets and multi-layer feed-forward nets, and conventional techniques such as, discriminant analysis, probit analysis and logistic regression, in evaluating credit risk in Egyptian banks applying credit scoring models. This work concluded that neural network models gave better average correct classification rates than the other techniques. However, in their neural network training and testing strategy, they used a high ratio of the dataset for training (80%), in comparison to validation (20%); which we consider as an imbalanced strategy when attempting to achieve meaningful neural network learning. In (Angelini, Di Tollo, & Roli, 2008), an application of neural networks to credit risk assessment related to Italian small businesses was described. This work presented two neural network systems, one with a standard feed-forward network, while the other with a special purpose architecture; and suggested that both neural networks can be very successful in learning and estimating the default tendency of a borrower, provided that careful data analysis, data pre-processing and training are performed.

In (Yu et al., 2008), a multistage neural network ensemble learning model was proposed to evaluate credit risk at the measurement level. The proposed model consisted of six stages: firstly, generating different training data subsets especially for data shortage, secondly, creating different neural network models with different training subsets obtained from the previous stage, thirdly, training the generated neural network models with different training datasets and obtaining the classification score, fourthly, selecting the appropriate ensemble members, fifthly, selecting the reliability values of the selected neural network models, and finally fusing the selected neural network ensemble members to obtain final classification result by means of reliability measurement. In (Tsai & Wu, 2008), the work investigated the performance of a single classifier as the baseline classifier to compare with multiple classifiers and diversified multiple classifiers by using neural networks based on three datasets. In (Setiono, Baesens, & Mues, 2008), a recursive algorithm for extracting classification rules from feed-forward neural networks that have been trained on credit scoring data sets; having both discrete and continuous attributes, was presented. Lately, in (Šušteršic, Mramor, & Zupan, 2009), Kohonen and error back-propagation neural networks were used as consumer credit scoring models for financial institutions where data usually used in previous research is not available. This work suggested that the error back-propagation neural network showed the best results. In (Lin, 2009), a three two-stage hybrid models of logistic regression-artificial neural network was proposed to construct a financial distress warning system suitable for Taiwan's banking industry, and to provide an optimal model of credit risk for supervising authorities, analysts and practitioners in conducting risk assessment and decision making. In (Wang & Huang, 2009), a back propagation based neural network was used to classify credit applicants. In (Xu, Zhou, & Wang, 2009), a credit scoring algorithm-based on support vector machines, was proposed to decide whether a bank should provide a loan to the applicant.

In (Chuang & Lin, 2009), a reassigning credit scoring model (RCSM) involving two-stages was proposed. The classification stage is constructing an ANN-based credit scoring model, which classifies applicants with accepted (good) or rejected (bad) credits. The reassign stage is trying to reduce the Type I error by reassigning the rejected good credit applicants to the conditional accepted class by using the CBR-based classification technique.

In general, we can deduce that using neural networks for credit scoring and evaluation has been shown to be effective over the past decade. The capability of neural networks in such applications is due to the way the network operates, and the availability of training data. This is more evident when using multi-layer perceptron networks based on the back propagation learning algorithm (Haykin, 1999). When feeding the information from a credit applicant to the neural network, attributes (applicant's answers to a set of questions (or characteristics)) are taken as the input to the neural network and a linear combination of them is taken with arbitrary weights. The attributes are linearly combined and subject to a non-linear transformation represented by a certain activation function (sigmoid function in this work), then fed as inputs into the next layer for similar manipulation. The final function yields values which can be compared with a cut-off for classification. Each training case is submitted to the network, the final output compared with the observed value and the difference, the error, is propagated back though the network and the weights modified at each layer according to the contribution each weight makes to the error value (Crook, Edelman, & Thomas, 2007). In essence the network takes data in attributes space, transforms it using the weights and activation functions into hidden value space and then possibly into further hidden value space; if further layers exist, and eventually into output layer space which is linearly separable.

Despite their successful application to credit scoring and evaluation, neural networks may not deliver robust "judgment" on whether an applicant should be granted credit or not. This problem arises from different reasons and partly depends on the chosen real world dataset for training and validating the trained neural network. Many of the previous works, which we described earlier on in this paper, suffer from problems despite the demonstrated successful implementations of the neural networks.

The first problem when using neural networks is the use of a high ratio of training-to-validation datasets. Depending on which dataset is used (the German credit dataset (Asuncion & Newman, 2007) is used in this work), a high ratio of training-to-validation data does not yield meaningful learning; for example, previously adopted ratios of training-to-validation (training:validation) datasets include: 80%:20% (Abdou, Pointon, & El-Masry, 2008; Li et al., 2006), 71%:29% (Boros et al., 2000), 68%:32% (Hsieh, 2005), 70%:30% (Baesens et al., 2003b; Hsieh, 2005; Kim & Sohn, 2004; Tsai & Wu, 2008), 69%:31% (Šušteršic et al., 2009), 67%:33.3% (Setiono et al., 2008), and 62%:38% (Atiya, 2001). A more appropriate ratio would be closer to (50%:50%) as used in (Sakprasat & Sinclair, 2007), or a lower ratio of training-to-testing dataset.

The second problem with using neural networks for credit evaluation is normalization of the input data. The values fed to the input layer of a neural network are usually between '0' to '1'. This is not a problem when using a neural network for image processing for example, since all input values would be representing the image pixel values, which in turn have a more uniform distribution and a finite difference between the lowest and the largest pixel value (Khashman, 2008). However, with credit evaluation, the numerical values (input values) representing the attributes of a credit applicant vary marginally in value, and if a simple normalization process is applied to the whole dataset, say by dividing each value in the set by the largest recorded value, then much information would be lost across the different attributes. For example, the highest value recorded in the German dataset is 184 (case 916, attribute 4); if all values within the dataset are divided by this maximum value, much of the input data would be closer to '0' value, which does not represent the attributes, thus leading to inefficient neural network training. Therefore, normalization of the credit application input data should be carefully performed, while maintaining the meaning of each attribute.

Another problem with using neural networks in financial applications is the computational cost. The simplest MLP neural network has three-layers (input, hidden and output). Much of the previously suggested neural networks for credit evaluation use two hidden layers. The problem here is the more layers are added, the higher the computational cost is, and thus, the higher the processing time becomes.

In this paper, we aim to address the above problems when designing neural network models with application to credit risk evaluation. Firstly, using the German credit dataset (Asuncion & Newman, 2007) that contains 1000 real world application–decision cases, we train three neural network models using nine learning schemes. The three neural models differ in topology, and in particular in the number of hidden layer neurons and learning and momentum rates. The nine learning schemes differ in the training-to-validation data ratios (or as we refer to them, learning ratios). The lower the ratio, the more challenging it is for a neural network, but the more robust and meaningful the learning is. We compare the performance of the neural network models under all schemes and then select the ideal neural model and learning scheme.

Secondly, we use a simple but efficient normalization procedure that is applied automatically when reading the input data for each numerical attribute value separately. This assures that the 24 input values representing the different 20 attributes are meaningful for the neural network after normalization. Thirdly, we maintain simplicity when designing the back propagation learning algorithm-based neural networks, by using a single hidden layer, and a single neuron at the output layer; thus minimizing the computational and time costs.

The structure of the paper is as follows: in Section 2 a brief explanation of the credit risk evaluation dataset is presented. In Section 3 the credit evaluation system is described; showing input data normalization procedure and the design strategy of the neural network models. In Section 4 the results of training and testing (validating) the neural models using the nine learning schemes are presented; and a comparison between the evaluation results is provided. Finally, Section 5 concludes this work and suggests future work.

2. Dataset for credit evaluation

For the implementation of our proposed credit evaluation system we use the German credit dataset; available publicly at UCI Machine Learning data repository (Asuncion & Newman, 2007).

Table 1
Decision attributes used for evaluating credit risk in the German credit dataset (Asuncion & Newman, 2007).

Attribute	Description	Class
1.	Status of existing checking account	Categorical
2.	Duration in months	Numerical
3.	Credit history	Categorical
4.	Purpose	Categorical
5.	Credit account	Numerical
6.	Savings account/bonds	Categorical
7.	Present employment since	Categorical
8.	Installment rate in percentage of disposable income	Numerical
9.	Personal status and sex	Categorical
10.	Other debtors/guarantors	Categorical
11.	Present residence since	Numerical
12.	Property	Categorical
13.	Age in years	Numerical
14.	Other installment plans	Categorical
15.	Housing	Categorical
16.	Number of existing credits at this bank	Numerical
17.	Job	Categorical
18.	Number of people being liable to provide maintenance for	Numerical
19.	Have telephone or not	Categorical
20.	Foreign worker	Categorical

This real world dataset, which classifies credit applicants described by a set of attributes as good or bad credit risks, has been successfully used for credit scoring and evaluation systems in many previous works (Eggermont, Kok, & Kosters, 2004; Huang, Tzeng, & Ong, 2006; Huang et al., 2007; Laha, 2007; Li et al., 2006; O'Dea, Griffith, & O'Riordan, 2001; Piramuthu, 2006; Setiono et al., 2008; Tsai & Wu, 2008; Yu, Wang, & Lai, 2009; Šušteršic et al., 2009).

The German credit dataset contains 1000 instances or cases of loan applications. The original data has a mix of 20 categorical and numerical attributes (see Table 1); recording various financial and demographic information about the applicants. In the repository a numeric version of this dataset is also available where the categorical attributes are transformed into numerical ones and a few indicator variables are added, which increases the dimension to 24 input numerical values. The data instances are labeled as classes 1 (good, 700 instances) and 2 (bad, 300 instances).

Table 2 shows examples of the dataset attributes' numerical representation for the first 10 cases; these numerical values are not normalized. Once normalization to values between "0" and "1" is completed, the values are used as the input data to a neural network. The dataset is divided into two subsets in our work; the training dataset and the validation or testing dataset. We devised nine learning schemes with different training-to-validation data ratios in order investigate an ideal training-to-validation ratio for the implementation of the credit evaluation neural network system.

3. The evaluation system

The neural network-based credit risk evaluation system consists of two phases: a data processing phase where each numerical value of the applicant's attributes within the dataset is normalized separately; this is one of our objectives in this work. The output of this phase provides normalized numerical values representing a credit applicant's case, which is used in the second phase; evaluating the applicant's attributes and deciding whether to accept or reject the application using a neural network. Once the neural network converges to a set value of minimum error, learning is accomplished and the second phase consists only of one forward pass that yields the evaluation result.

Table 2
Examples of pre-normalization numerical input values representing the input/output attributes for the first 10 cases in the German credit dataset (Asuncion & Newman, 2007). Output 1 indicates accept application; while output 2 indicates reject application.

	Numerical values of input attributes																								Output
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
1.	1	6	4	12	5	5	3	4	1	67	3	2	1	2	1	0	0	1	0	0	1	0	0	1	1
2.	2	48	2	60	1	3	2	2	1	22	3	1	1	1	1	0	0	1	0	0	1	0	0	1	2
3.	4	12	4	21	1	4	3	3	1	49	3	1	2	1	1	0	0	1	0	0	1	0	1	0	1
4.	1	42	2	79	1	4	3	4	2	45	3	1	2	1	1	0	0	0	0	0	0	0	0	1	1
5.	1	24	3	49	1	3	3	4	4	53	3	2	2	1	1	1	0	1	0	0	0	0	0	1	2
6.	4	36	2	91	5	3	3	4	4	35	3	1	2	2	1	0	0	1	0	0	0	0	1	0	1
7.	4	24	2	28	3	5	3	4	2	53	3	1	1	1	1	0	0	1	0	0	1	0	0	1	1
8.	2	36	2	69	1	3	3	2	3	35	3	1	1	2	1	0	1	1	0	1	0	0	0	0	1
9.	4	12	2	31	4	4	1	4	1	61	3	1	1	1	1	0	0	1	0	0	1	0	1	0	1
10.	2	30	4	52	1	1	4	2	3	28	3	2	1	1	1	1	0	1	0	0	1	0	0	0	2

3.1. Credit application data processing

This phase is a data preparation phase for neural network training and classification/evaluation. Here, the input data (attribute numerical values) are separately normalized to values between "0" and "1". This is achieved by finding the maximum or highest value within each input attribute for all 1000 instances or cases in the dataset, and dividing all the values within that same attribute by the obtained maximum value. Table 3 shows the maximum values for each input attribute. The highest value in the whole set belongs to case 916, numerical attribute input 4 (Max = 184) which is a large value in comparison to the other values in the dataset, and thus, if used for normalizing the remaining input data values, it would lead to normalized values closer to "0". This could cause insufficient learning of the neural network, as the applicant's attributes would not be appropriately represented. Table 4 shows the normalized input values (representing the attributes) of the first 5 cases in the dataset.

3.2. Neural network arbitration

During this phase we use a supervised neural network that is based on the back propagation learning algorithm due to its implementation simplicity and the availability of sufficient dataset for training and validating this supervised learner. Fig. 1 shows the general topology of the credit evaluation neural network model.

The neural network input layer has 24 neurons, according to the number of the applicant's attributes numerical input values; each input neuron receives a normalized numerical value. There is one hidden layer containing h neurons; the number of hidden neuron

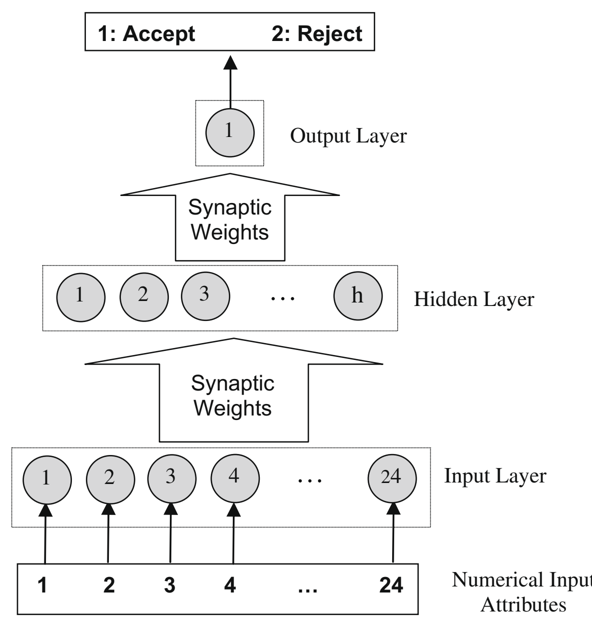

Fig. 1. General topology of credit risk evaluation neural network model for neural network models: ANN-1 (h = 18), ANN-2 (h = 23), and ANN-3 (h = 27).

Table 3
The highest or maximum value in each numerical input of the 1000 credit application cases; these values are used to normalize the input data prior to neural network arbitration.

Input	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Maximum value	4	72	4	184	5	5	4	4	4	75	3	4	2	2	2	1	1	1	1	1	1	1	1	1

Table 4
Normalized input data-attribute numerical values for the first 5 cases in the credit dataset.

Input	1	2	3	4	5	6	7	8	9	10	11	12
Case 1	0.250000	0.083333	1.000000	0.065217	1.000000	1.000000	0.750000	1.000000	0.250000	0.893333	1.000000	0.500000
Case 2	0.500000	0.666667	0.500000	0.326087	0.200000	0.600000	0.500000	0.500000	0.250000	0.293333	1.000000	0.250000
Case 3	1.000000	0.166667	1.000000	0.114130	0.200000	0.800000	0.750000	0.750000	0.250000	0.653333	1.000000	0.250000
Case 4	0.250000	0.583333	0.500000	0.429348	0.200000	0.800000	0.750000	1.000000	0.500000	0.600000	1.000000	0.250000
Case 5	0.250000	0.333333	0.750000	0.266304	0.200000	0.600000	0.750000	1.000000	1.000000	0.706667	1.000000	0.500000
Input	13	14	15	16	17	18	19	20	21	22	23	24
Case 1	0.500000	1.000000	0.500000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000
Case 2	0.500000	0.500000	0.500000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000
Case 3	1.000000	0.500000	0.500000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	1.000000	0.000000
Case 4	1.000000	0.500000	0.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
Case 5	1.000000	0.500000	0.500000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000

depends on the neural network model. We implement our investigation using three neural models: ANN-1 with $h = 18$, ANN-2 with $h = 23$ and ANN-3 with $h = 27$. The optimum number of hidden neurons h in all three models, which assures meaningful training while keeping the time cost to a minimum, was obtained after several experiments involving the adjustment of the number of hidden neurons from one to 50 neurons. The output layer has one single neuron, which uses binary output data representation; '0' for accepting or '1' for rejecting a credit application. Notice that the output classification in the German credit dataset uses "1" and "2" for "accept" and "reject", respectively. We have simply re-coded the output classification to binary 0 and 1. A simple thresholding scheme is then sufficient for the neural network's single output neuron to divide the feature space into the two categories. A threshold value of 0.5 is used to distinguish between credit groups, good credit and bad credit. If the output result of the neural network is greater than or equal to 0.5, the presented case is assigned to one class (good, accept); otherwise it is assigned to the other class (bad, reject). Hence,

$$\begin{aligned} &Applicant_i \in Good\ Credit\ Class\ if: \quad NN_{out}(i) \geqslant 0.5 \\ &Applicant_i \in Bad\ Credit\ Class\ if: \quad NN_{out}(i) < 0.5, \end{aligned} \tag{1}$$

where $NN_{out}(i)$ is the output of the neural network model obtained when the attributes of the ith case (applicant) are presented to the network. This is basically the output credit decision associated with applicant i.

Under nine different learning schemes the learning coefficient and the momentum rate of each of the three neural network models were adjusted during various experiments in order to achieve the required minimum error value of 0.008 which was considered as sufficient for this application. Table 5 lists the final parameters of the trained neural network models.

Table 5
Neural network models final training parameters.

Neural network model	ANN-1	ANN-2	ANN-3
Input layer nodes	24	24	24
Hidden layer nodes	18	23	27
Output layer nodes	1	1	1
Learning coefficient	0.0081	0.0095	0.0075
Momentum rate	0.70	0.69	0.79
Random initial weight range	−0.3 to +0.3	−0.3 to +0.3	−0.3 to +0.3
Minimum required error	0.008	0.008	0.008
Obtained error	0.007972	0.008000	0.008531
Maximum allowed iterations	25,000	25,000	25,000
Performed iterations	19,236	18,652	25,000
Optimum training-to-validation ratio	500:500	400:600	600:400

4. Implementation and experimental results

The results of implementing the credit risk evaluation neural network models were obtained using a 2.8 GHz PC with 2 GB of RAM, Windows XP OS and Borland C++ compiler. As one of our objectives is to investigate an ideal learning ratio, we follow nine learning schemes to train the neural network.

The learning schemes differ in the training-to-validation data ratio. For example, learning scheme 1 (LS1) uses a ratio of (100:900); i.e. the first 100 credit application cases are used for training the network, while the remaining 900 cases are not exposed to the neural network during training, as they will be used to test or validate the network's classification upon completion

Table 6
The nine learning schemes and implementation results of the three credit risk evaluation neural network models. T-dataset: training dataset. V-dataset: validation dataset.

Learning scheme	Learning ratio	Neural network	Obtained error	Training time[a] (s)	Run time[a] (s)	T-dataset accuracy rate (%)	V-dataset accuracy rate (%)	Overall rate[b] (%)
LS 1	100:900	ANN-1	0.020177	80.95	10.3×10^{-5}	(98/100) 98	(639/900) 71	73.7
		ANN-2	0.020145	92.31	3.44×10^{-5}	(98/100) 98	(641/900) 71.20	73.9
		ANN-3	0.020122	102.11	10.3×10^{-5}	(98/100) 98	(642/900) 71.33	73.9
LS 2	200:800	ANN-1	0.021285	133.11	3.88×10^{-5}	(196/200) 98	(554/800) 69.25	75
		ANN-2	0.010136	145.81	7.75×10^{-5}	(198/200) 99	(561/800) 70.13	75.9
		ANN-3	0.015112	164.45	23.5×10^{-5}	(197/200) 98.5	(560/800) 70	75.7
LS 3	300:700	ANN-1	0.013566	161.67	2.14×10^{-5}	(296/300) 98.67	(507/700) 72.43	80.3
		ANN-2	0.010148	206.61	13.4×10^{-5}	(298/300) 99	(495/700) 70.71	79.3
		ANN-3	0.266667	222.95	15.6×10^{-5}	(220/300) 73.33	(480/700) 68.57	70
LS 4	**400:600**	ANN-1	0.012787	203.42	2.67×10^{-5}	(395/400) 98.75	(429/600) 71.5	82.4
		ANN-2	**0.008000**	**183.88**	$\mathbf{5.17 \times 10^{-5}}$	**(397/400) 99.25**	**(439/600) 73.17**	**83.6**
		ANN-3	0.270000	285	13×10^{-5}	(292/400) 73	(408/600) 68	70
LS 5	**500:500**	**ANN-1**	**0.007972**	**193.08**	$\mathbf{6.2 \times 10^{-5}}$	**(496/500) 99.2**	**(356/500) 71.2**	**85.2**
		ANN-2	0.272000	298.41	9.4×10^{-5}	(364/500) 72.8	(336/500) 67.2	70
		ANN-3	0.012155	344.45	15.6×10^{-5}	(494/500) 98.8	(338/500) 67.6	83.2
LS 6	**600:400**	ANN-1	0.015179	293.88	8×10^{-5}	(591/600) 98.5	(282/400) 70.5	87.3
		ANN-2	0.295000	358.88	11.75×10^{-5}	(423/600) 70.5	(277/400) 69.25	70
		ANN-3	**0.008531**	**414.16**	$\mathbf{7.75 \times 10^{-5}}$	**(595/600) 99.17**	**(264/400) 66**	**85.9**
LS 7	700:300	ANN-1	0.017404	338.22	10.67×10^{-5}	(688/700) 98.29	(216/300) 72	90.4
		ANN-2	0.295714	454.09	21×10^{-5}	(493/700) 70.43	(207/300) 69	70
		ANN-3	0.015823	468.38	20.67×10^{-5}	(689/700) 98.43	(224/300) 74.67	91.3
LS 8	800:200	ANN-1	0.028982	376.58	7.5×10^{-5}	(777/800) 97.13	(149/200) 74.5	92.6
		ANN-2	0.298750	487.56	23.5×10^{-5}	(561/800) 70.13	(139/200) 69.5	70
		ANN-3	0.025132	536.39	31.5×10^{-5}	(780/800) 97.5	(147/200) 73.5	92.7
LS 9	900:100	ANN-1	0.297778	410.38	16×10^{-5}	(632/900) 70.22	(68/100) 68	70
		ANN-2	0.297778	510.72	31×10^{-5}	(632/900) 70.22	(68/100) 68	70
		ANN-3	0.297778	579.09	31×10^{-5}	(632/900) 70.22	(68/100) 68	70

[a] Using a 2.8 GHz PC with 2 GB of RAM, Windows XP Operating System and C++ programming language.
[b] Overall accuracy rate is obtained by combining accuracy rates of training and validation datasets.

of training. The learning ratios for the nine learning schemes are shown in Table 6.

To compare the performance of the three neural network models under the nine learning schemes, performance evaluation criteria are required. The criteria were firstly based on training the neural models under all schemes for a chosen number of iterations; 25,000 iterations were considered as sufficient, since a higher number of iterations would lead to increasing the computational costs. The second criterion was based on the network converging to the required error value of 0.008. Upon achieving either of these criteria, training the neural model is terminated and the time costs, in addition to the correct evaluation rates (accuracy rates) are recorded. Table 6 describes in details the obtained results for the three neural models after training under the nine learning schemes.

Upon inspecting the results in Table 6, we notice that only three implementations satisfy, or closely satisfy our evaluation criteria. Firstly, the implementation of the proposed credit evaluation system using neural model ANN-2 under LS4 learning scheme (learning ratio: 400:600) satisfied both criteria by requiring the least number of iterations (18,652 iterations) to converge to the required error value of 0.008. Secondly, the implementation using neural model ANN-1 under LS5 learning scheme (learning ratio: 500:500) also satisfied both criteria; albeit with a higher number of iterations (19,236 iterations). Thirdly, the implementation using neural model ANN-3 under LS6 learning scheme (learning ratio: 600:400) closely satisfied both criteria; requiring 25,000 iterations to converge to error value of 0.008531. The rest of the 27 implementations did not satisfy either criterion. Thus, we focus on the aforementioned three successful implementations, and compare their computational costs and accuracy rates. Fig. 2 shows the learning curves for the three successfully trained neural network models.

When considering the computational costs, the first successful implementation (LS4, ANN-2) required approximately 184 s for training, which is less than the other two successful implementations. This, however, is anticipated since the number cases in the training dataset, is less than the other two implementations. Considering the run time, which is basically the time taken by a trained neural model to perform one forward pass and decide whether to accept a credit application or not, again the first implementation (LS4, ANN-2) required the least time; making a decision on a credit application case within 5.17×10^{-5} s. Therefore, in terms of computational cost the first implementation using neural model ANN-2 and learning scheme LS4, is the most computationally cost-effective.

More importantly is the accuracy rate of the credit evaluation system. At first impression, and looking at the overall accuracy rates, the third implementation (ANN-3, LS6) appears to have the highest overall rate of 85.9%. However, careful analysis of the obtained results shows that this overall rate can be misleading. A more appropriate comparison requires inspecting the accuracy rates of the training dataset (T-dataset) and the validation dataset (V-dataset) separately. This is because the V-dataset accuracy rate is obtained by exposing the trained neural model to unseen inputs or cases, thus reflecting the robustness of the trained model. The T-dataset accuracy rate is also significant in particular with the German credit dataset, which is considered as unbalanced and difficult to process by intelligent systems. Based on these considerations, further investigation of the obtained results, reveals that the highest accuracy rates amongst the successfully trained models belong to neural model ANN-2, under learning scheme LS4; achieving 99.25% T-dataset accuracy rate, and 73.17% V-dataset accuracy rate. When combining the observed results, we find out that the credit risk evaluation system can be successfully and efficiently implemented, with an optimum configuration when using neural network model ANN-2 trained under learning scheme LS4, i.e. with a training-to-validation ratio of 40%:60%.

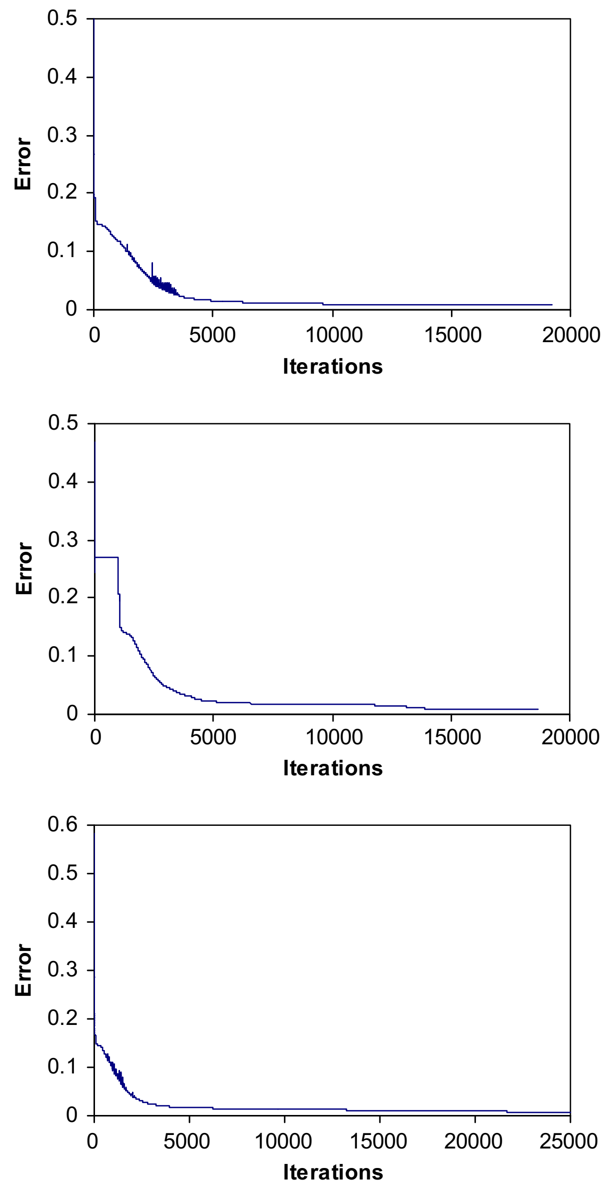

Fig. 2. Learning curves (error convergence) of neural network models: (a) ANN-1 under learning scheme LS5, (b) ANN-2 under learning scheme LS4 and (c) ANN-3 under learning scheme LS6.

5. Conclusions

This paper presented an investigation of the use of supervised neural network models for credit risk evaluation under different learning schemes. We also propose an efficient, fast and simple to use credit evaluation system, based on the results of our investigation. In our approach we trained three models of a three-layer supervised neural network; based on the back propagation learning algorithm, under nine learning schemes. These schemes differ in the ratio of the number of credit application cases used for training, against those used for validation or testing. The use of high training-to-validation dataset ratio has been a common problem with much of the previous research works on using neural networks for credit risk evaluation as shown in Section 1. We also described in detail (Section 4) the criteria and considerations which are to be made in order to decide upon an optimum learning scheme and neural model.

Furthermore, we proposed a simple but efficient method of normalizing the input data (attributes of a credit applicant) prior to presenting it to the neural network. We also maintain simplicity in designing the neural network model, in order to keep the computational and time costs to a minimum, thus, the use of one hidden layer; many previous works generally use two hidden layers for similar applications.

In order to implement our proposed neural system for credit risk evaluation, we used the German credit dataset. This is a real world dataset that has been successfully used for credit scoring and evaluation systems in previous works. The German credit data consists of 1000 cases or instances: 700 instances of creditworthy applicants (“good” or “accept”), and 300 instances where credit is not creditworthy (“bad” or “reject”).

In conclusion, the credit risk evaluation neural network model performs best when using the LS4 learning scheme; which uses 400 cases for training and 600 cases for validation (40%:60%). Accuracy rates of 99.25% and 73.17% were obtained using the training and validation data sets, respectively; the overall accuracy rate was 83.6%. Training this neural model was completed in approximately 184 s, whereas the decision making time for the trained neural model was a fast 5.17×10^{-5} s. Therefore, we suggest that this neural system can be efficiently used in automatic processing of credit applications. Future work will focus on designing, training and implementing a neural system with more outputs, which could indicate the reason why a credit application had been rejected.

References

Abdou, H., Pointon, J., & Elmasry, A. (2008). Neural nets versus conventional techniques in credit scoring in Egyptian banking. *Expert Systems and Applications, 35*(3), 1275–1292.

Ahn, B. S., Cho, S. S., & Kim, C. Y. (2000). The integrated methodology of rough set theory and artificial neural network for business failure prediction. *Expert Systems with Applications, 18*(2), 65–74.

Angelini, E., Di Tollo, G., & Roli, A. (2008). A neural network approach for credit risk evaluation. *The Quarterly Review of Economics and Finance, 48*, 733–755.

Asuncion, A., & Newman, D. J. (2007). *UCI machine learning repository*. Irvine, CA: University of California, School of Information and Computer Science <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Atiya, A. F. (2001). Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Transactions on Neural Networks, 12*(4), 929–935.

Baesens, B., Sentiono, R., Mues, C., & Vanthienen, J. (2003a). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science, 49*(3), 312–329.

Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003b). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of Operational Research, 54*, 627–635.

Baesens, B., Gestel, T. V., Stepanova, M., Van den Poel, D., & Vanthienen, J. (2005). Neural network survival analysis for personal loan data. *Journal of the Operational Research Society, 56*(9), 1089–1098.

Basel Committee on Banking Supervision (2000). *Principles for the management of credit risk* (p. 11). Basel, Switzerland <http://www.bis.org/publ/bcbs75.pdf?noframes=1>.

Basel Committee on Banking Supervision (2005). *International convergence of capital management and capital standards: A revised framework* (p. 15). Basel, Switzerland <http://www.bis.org/publ/bcbs118.htm>.

Becerra-Fernandez, I., Zanakis, S. H., & Walczak, S. (2002). Knowledge discovery techniques for predicting country investment risk. *Computers and Industrial Engineering, 43*(4), 787–800.

Boros, E., Hammer, P. L., Ibaraki, T., Kogan, A., Mayoraz, E., & Muchnik, I. (2000). An implementation of logical analysis of data. *IEEE Transactions on Knowledge and Data Engineering, 12*(2), 292–306.

Cheng, E. W. L., Chiang, Y. H., & Tang, B. S. (2007). Alternative approach to credit scoring by DEA: Evaluating borrowers with respect to PFI projects. *Building and Environment, 42*, 1752–1760.

Chuang, C. L., & Lin, R. H. (2009). Constructing a reassigning credit scoring model. *Expert Systems with Applications, 36*(2), 1685–1694.

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research, 183*, 1447–1465.

Eggermont, J., Kok, J. N., & Kosters, W. A. (2004). Genetic programming for data classification: Partitioning the search space. In *Proceedings of the 2004 symposium on applied computing* (pp. 1001–1005). Cyprus.

Haykin, S. (1999). *Neural networks. A comprehensive foundation* (2nd ed.). Prentice-Hall.

Hsieh, N. C. (2005). Hybrid mining approach in design of credit scoring model. *Expert Systems with Applications, 28*, 655–665.

Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*(4), 847–856.

Huang, J. J., Tzeng, G. H., & Ong, C. S. (2006). Two-stage genetic programming (2SGP) for the credit scoring model. *Applied Mathematics and Computation, 174*, 1039–1053.

Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers and Operations Research, 32*(10), 2513–2522.

Huang, Z., Chen, H., Hsu, C. J., Chen, W. H., & Wu, S. (2004). Credit rating analysis with support vector machines and neural networks: A market comparative study. *Decision Support Systems, 37*(4), 543–558.

Khashman, A. (2008). Blood cell identification using a simple neural network. *International Journal of Neural Systems, 18*(5), 453–458.

Kim, Y. S., & Sohn, S. Y. (2004). Managing loan customers using misclassification patterns of credit scoring model. *Expert Systems with Applications, 26*, 567–573.

Laha, A. (2007). Building contextual classifiers by integrating fuzzy rule based classification technique and *k*-nn method for credit scoring. *Advanced Engineering Informatics, 21*, 281–291.

Lee, T. S., & Chen, I. F. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications, 28*, 743–752.

Lee, T. S., Chiu, C. C., & Lu, C. J. (2002). Credit scoring using the hybrid neural discriminant technique. *Expert Systems with Application, 23*(3), 245–254.

Li, S. T., Shiue, W., & Huang, M. H. (2006). The evaluation of consumer loans using support vector machines. *Expert Systems with Applications, 30*(4), 772–782.

Lim, M. K., & Sohn, S. Y. (2007). Cluster-based dynamic scoring model. *Expert Systems with Applications, 32*(2), 427–431.

Lin, S. L. (2009). A new two-stage hybrid approach of credit risk in banking industry. *Expert Systems with Applications, 36*(4), 8333–8341.

Malhotra, R., & Malhotra, D. K. (2003). Evaluating consumer loans using neural networks. *Omega, 31*(2), 83–96.

Martens, D., Baesens, B., Van Gestel, T., & & Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research, 183*, 1466–1476.

Min, J. H., & Lee, Y. C. (2005). Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Systems with Applications, 28*(4), 603–614.

Min, J. H., & Lee, Y. C. (2008). A practical approach to credit scoring. *Expert Systems with Applications, 35*(4), 1762–1770.

O'Dea, P., Griffith, J., & O'Riordan, C. (2001). *Combining feature selection and neural networks for solving classification problems*. Technical Report of the Dept of IT, NUI Galway, Number NUIG-IT-130601.

Piramuthu, S. (2006). On preprocessing data for financial credit risk evaluation. *Expert Systems with Applications, 30*(3), 489–497.

Sakprasat, S., & Sinclair, M. C. (2007). Classification rule mining for automatic credit approval using genetic programming. In *Proceedings of the IEEE congress on evolutionary computation* (pp. 548–555). Singapore.

Setiono, R., Baesens, B., & Mues, C. (2008). Recursive neural network rule extraction for data with mixed attributes. *IEEE Transactions on Neural Networks, 19*(2), 299–307.

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

Šušteršic, M., Mramor, D., & Zupan, J. (2009). Consumer credit scoring models with limited data. *Expert Systems with Applications, 36*(3), 4736–4744.

Tsai, C. F., & Wu, J. W. (2008). Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 34*(4), 2639–2649.

Vellido, A., Lisboa, P. J. G., & Vaughan, J. (1999). Neural networks in business: A survey of applications (1992–1998). *Expert Systems with Applications, 17*(1), 51–70.

Wang, C. M., & Huang, Y. F. (2009). Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data. *Expert Systems with Applications, 36*(3), 5900–5908.

West, D., Dellana, S., & Qian, J. (2005). Neural network ensemble strategies for financial decision applications. *Computers and Operations Research, 32*(10), 2543–2559.

Xu, X., Zhou, C., & Wang, Z. (2009). Credit scoring algorithm based on link analysis ranking with support vector machine. *Expert Systems with Applications, 36*(2), 2625–2632.

Yu, L., Wang, S. Y., & Lai, K. K. (2008). Credit risk assessment with a multistage neural network ensemble learning approach. *Expert Systems with Applications, 34*(2), 1434–1444.

Yu, L., Wang, S. Y., & Lai, K. K. (2009). An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: The case of credit scoring. *European Journal of Operational Research, 195*, 942–959.