

A NEURAL NETWORK MODEL FOR CREDIT RISK EVALUATION

ADNAN KHASHMAN

*Intelligent Systems Research Group (ISRG)
Near East University, Lefkosa, Mersin 10, Turkey
khashman@ieee.org*

Credit scoring is one of the key analytical techniques in credit risk evaluation which has been an active research area in financial risk management. This paper presents a credit risk evaluation system that uses a neural network model based on the back propagation learning algorithm. We train and implement the neural network to decide whether to approve or reject a credit application, using seven learning schemes and real world credit applications from the Australian credit approval datasets. A comparison of the system performance under the different learning schemes is provided, furthermore, we compare the performance of two neural networks; with one and two hidden layers following the ideal learning scheme. Experimental results suggest that neural networks can be effectively used in automatic processing of credit applications.

Keywords: Credit scoring and evaluation; personal bank loan; neural network; Australian credit approval.

1. Introduction

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. Credit risk analysis is an important topic in the financial risk management, and has been the major focus of financial and banking industry.¹ Data mining methods, especially pattern classification, using real-world historical data, is of paramount importance in building such predictive models. Credit scoring tasks can be divided into two distinct types.^{2–4} 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. In this paper, we shall focus on application scoring.

In the past few decades, artificial neural networks have been successfully applied in various fields,^{5–13} as well as the financial field. The use of neural

networks has become very popular in various business applications including credit scoring. In contrast to the statistical methods, in neural network-based techniques one need not make assumptions regarding the distribution of the data or find it through estimation techniques.¹⁴

The use of neural networks in business applications has been previously investigated by several works.^{4,15–27} The general outcome of these works is that in the credit industry, neural networks have been considered to be an accurate tool for credit analysis among others.^{21–31}

Recently, the work in Ref. 32 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 inefficient training strategy when attempting to achieve meaningful neural network learning. In Ref. 33, two neural network systems were proposed for credit risk assessment related to Italian small businesses. In Ref. 34, a multistage neural network ensemble learning model was proposed to evaluate credit risk at the measurement level. In Ref. 35, 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 Ref. 36, 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 Ref. 37, 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.

Although the use of neural networks for credit scoring and evaluation has been shown to be successful over the past decade, there are few problems with the way these neural networks are trained, and this may affect the robustness of the neural system's judgment on whether an applicant should be granted credit or not.

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 Australian credit dataset³⁸ 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%^{2,32}), (71%:29%³⁹), (68%:32%²⁴), (70%:30%^{24,35,40,41}), (69%:31%³⁷), (67%:33.3%³⁶), and (62%:38%⁴²). We consider this as a problem in our opinion, since the neural network's task is made easier by exposing it to more data during training, thus reducing the significance of its generalization when tested or validated with the remaining data. A more appropriate ratio would be closer to (50%:50%) as used in Ref. 43, 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.^{44,45} 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 Australian dataset is 100001 (case 501, attribute 14), 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. Normalization of the input data is performed during both training and after training.

Another problem with using neural networks in financial applications is the computational cost. The simplest multilayer perceptron 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 a neural network model with application to credit risk evaluation.

Firstly, using the Australian credit dataset³⁸ that contains 690 real world application-decision cases, we train neural network model based on the BP learning algorithm using seven learning schemes, which differ in the training-to-validation data ratios. The lower the ratio, the more challenging it is for the neural network, but the more robust and meaningful the learning is. We compare the performance of the neural network model under all schemes and then select the ideal learning scheme.

Secondly, we use a normalization procedure that is applied automatically when reading the

input data for each attribute separately. This assures that the values representing the different 14 attributes are meaningful for the neural network after normalization.

Thirdly, we maintain simplicity when designing the back propagation learning algorithm-based neural network, by using a single hidden layer, and a single neuron at the output layer, thus minimizing the computational and time costs. We also compare the correct evaluation rate and the computational costs between a single-hidden layer neural network (SHNN) and a double-hidden layer neural network (DHNN).

The structure of the paper is as follows: in Sec. 2 a brief explanation of the Australian credit dataset is presented. In Sec. 3 the neural network system for credit evaluation is described, showing input data normalization procedure and the neural network design. In Sec. 4 the results of training and testing (validating) the neural network using the seven learning schemes are presented, and a comparison between the evaluation results under the seven learning schemes is provided. Moreover, the performance of the SHNN and DHNN models when trained under the chosen ideal scheme is compared. Finally, Sec. 5 concludes this work and suggests future work.

2. Credit Dataset

For the implementation of our proposed credit evaluation system we use the Australian credit dataset,³⁸ available publicly at UCI Machine Learning data repository. This is a real world dataset that has been

successfully used for credit scoring and evaluation systems in many previous works.^{2,30,31,35,37,39,43,46}

The Australian credit data consists of 307 instances of creditworthy applicants (“good”, “accept”) and 383 instances where credit is not creditworthy (“bad”, “reject”). Each instance or case contains six nominal, eight numeric attributes (inputs), and one class attribute (output: accept or reject). This dataset is interesting because there is a good mixture of attributes: continuous, nominal with small numbers of values, and nominal with larger numbers of values. The details of the attributes are available at the repository.³⁸ Table 1 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 the neural network.

The dataset is divided into two sub-sets in our work, the training dataset and the validation or testing dataset. We devised seven 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. Credit Risk Evaluation System

The neural network-based credit risk evaluation system consists of two phases: a data normalization phase where each attribute within the dataset is normalized separately, this is one of our objectives in this work.

Table 1. Examples of pre-normalization numerical values representing input/output attributes for the first 10 cases in the Australian credit dataset.³⁸

Cases	Input attributes														Output
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
1	1	22.08	11.46	2	4	4	1.585	0	0	0	1	2	100	1213	0
2	0	22.67	7	2	8	4	0.165	0	0	0	0	2	160	1	0
3	0	29.58	1.75	1	4	4	1.25	0	0	0	1	2	280	1	0
4	0	21.67	11.5	1	5	3	0	1	1	11	1	2	0	1	1
5	1	20.17	8.17	2	6	4	1.96	1	1	14	0	2	60	159	1
6	0	15.83	0.585	2	8	8	1.5	1	1	2	0	2	100	1	1
7	1	17.42	6.5	2	3	4	0.125	0	0	0	0	2	60	101	0
8	0	58.67	4.46	2	11	8	3.04	1	1	6	0	2	43	561	1
9	1	27.83	1	1	2	8	3	0	0	0	0	2	176	538	0
10	0	55.75	7.08	2	4	8	6.75	1	1	3	1	2	100	51	0

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 and learning is accomplished, the second phase consists only of one forward pass that yields the evaluation result.

3.1. Input data normalization

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 attribute for all 690 cases in the dataset, and dividing all the values within that same attribute by the maximum. Table 2 shows the maximum values for each input attribute. The highest value in the whole set belongs to attribute A14 (100001) which is a large value in comparison to the other values in the set, and thus, if used for normalizing the remaining input data values, it would lead to normalized values close to "0". This could cause insufficient learning of the neural network, as the applicant's attributes would not be appropriately represented. Table 3 shows the normalized

input values (attributes) of the first 10 cases in the dataset.

3.2. Neural network arbitration

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

The neural network input layer has 14 neurons, according to the number of the applicant's attributes; each input neuron receives a normalized numerical value representing an attribute. There is one hidden layer containing 9 neurons, this number, which was chosen after several experiments involving the adjustment of the number of hidden neurons from one to 80 neurons, assures meaningful training while keeping the time cost to a minimum. The output layer has one single neuron, which uses binary output data representation, "1" for accepting or "0" for rejecting a credit application. A simple thresholding scheme is sufficient for the neural network's single output neuron to divide the feature space into 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

Table 2. The highest value in each attribute of the 690 cases in the dataset. These values are used to normalize the input data prior to neural network processing.

Input attribute	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
Maximum value	1	80.25	28	3	14	9	28.5	1	1	67	1	3	2000	100001

Table 3. Normalized input data — attribute values for the first 10 cases in the Australian credit dataset-shown in Table 1.

		Normalized input attributes													
		A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
1	1	0.275140	0.409286	0.666667	0.285714	0.444444	0.055614	0	0	0.000000	1	0.666667	0.050000	0.012130	
2	0	0.282492	0.250000	0.666667	0.571429	0.444444	0.005789	0	0	0.000000	0	0.666667	0.080000	0.000010	
3	0	0.368598	0.062500	0.333333	0.285714	0.444444	0.043860	0	0	0.000000	1	0.666667	0.140000	0.000010	
4	0	0.270031	0.410714	0.333333	0.357143	0.333333	0.000000	1	1	0.164179	1	0.666667	0.000000	0.000010	
5	1	0.251340	0.291786	0.666667	0.428571	0.444444	0.068772	1	1	0.208955	0	0.666667	0.030000	0.001590	
6	0	0.197259	0.020893	0.666667	0.571429	0.888889	0.052632	1	1	0.029851	0	0.666667	0.050000	0.000010	
7	1	0.217072	0.232143	0.666667	0.214286	0.444444	0.004386	0	0	0.000000	0	0.666667	0.030000	0.001010	
8	0	0.731090	0.159286	0.666667	0.785714	0.888889	0.106667	1	1	0.089552	0	0.666667	0.021500	0.005610	
9	1	0.346791	0.035714	0.333333	0.142857	0.888889	0.105263	0	0	0.000000	0	0.666667	0.088000	0.005380	
10	0	0.694704	0.252857	0.666667	0.285714	0.888889	0.236842	1	1	0.044776	1	0.666667	0.050000	0.000510	

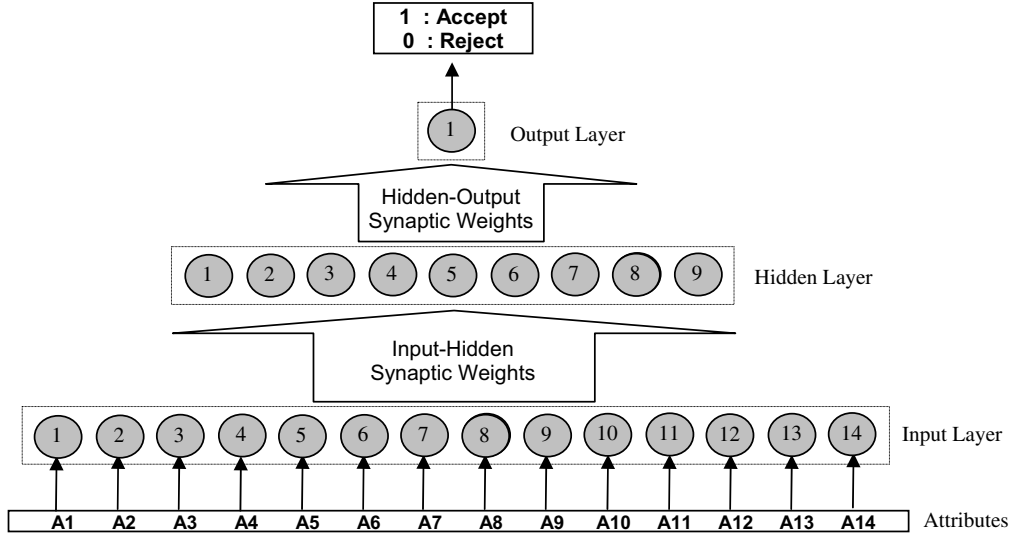


Fig. 1. Topology of the credit risk evaluation neural network model.

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,

$$\begin{aligned} \text{Applicant}_i &\in \text{Good Credit Class if : } NN_{out}(i) \geq 0.5 \\ \text{Applicant}_i &\in \text{Bad Credit Class if : } NN_{out}(i) < 0.5, \end{aligned} \quad (1)$$

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

During the learning phase, the learning coefficient, and the momentum rate were adjusted during various experiments in order to achieve the required minimum error value of 0.003 which was considered as sufficient for this application. The final training parameters for this neural model were: learning coefficient (0.00935) and momentum rate (0.67). The initial weights of the neural network were randomly generated with values between -0.3 and $+0.3$.

4. Implementation and Experimental Results

The results of implementing the credit risk evaluation neural network model were obtained using 2.8GHz PC with 2GB of RAM, Windows XP OS and Borland C++ compiler.

As one of our objectives is to investigate an ideal training-to-validation data ratio, we follow seven 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 (100:590), i.e., the first 100 credit application cases are used for training the network, while the remaining 590 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 of training. The training-to-validation ratios and their percentages for the seven learning schemes are shown in Tables 4 and 5. When choosing our different training-validation ratios (e.g., 100–590, 200–490, ... etc.), we simply start with the first 100 cases as training data and the remaining cases as testing or validation cases. For the next ratio, we add the next 100 cases to the training set and so on. There was one exception which was the ratio of (345:345) this was chosen as it provides 50% training cases and 50% validation/testing cases.

To compare the performance of the neural network model under the seven learning schemes, a performance evaluation criterion is required. The criteria was firstly based on training the neural network under all schemes for a chosen number of iterations, 50000 iterations was considered sufficient, as a higher number of iterations would lead to increasing the computational costs. The criteria was secondly based on the network converging to the required error value of 0.003, where the least number of

Table 4. Credit risk evaluation neural network's learning schemes and their implementation results after 20801 iterations.

Learning scheme	LS1	LS2	LS3	LS4	LS5	LS6	LS7
Training-to-validation ratio	100:590	200:490	300:390	345:345	390:300	400:290	500:190
Training-to-validation percent	14.5%–85.5%	29%–71%	43.5%–56.5%	50%–50%	56.5%–43.5%	58%–42%	72.5%–27.5%
Obtained error	0.011	0.020	0.003	0.018	0.011	0.023	0.022
Training time (seconds) ¹	56.89	60.98	73.92	81.59	87.67	93.28	100.39
Run time (seconds) ¹	2.12×10^{-4}	0.33×10^{-4}	0.38×10^{-4}	0.46×10^{-4}	0.53×10^{-4}	0.52×10^{-4}	0.84×10^{-4}
Training dataset correct evaluation rate	(99/100) 99%	(195/200) 97.50%	(300/300) 100%	(339/345) 98.26%	(385/390) 98.72%	(392/400) 98%	(489/500) 97.8%
Validation dataset correct evaluation rate	(273/590) 46.27%	(394/490) 80.41%	(316/390) 81.03%	(275/345) 79.71%	(237/300) 79%	(230/290) 79.31%	(152/190) 80%
Overall evaluation rate ²	(372/690) 53.91%	(589/690) 85.36%	(616/690) 89.28%	(614/690) 88.98%	(622/690) 90.14%	(622/690) 90.14%	(641/690) 92.90%

¹Using a 2.8 GHz PC with 2 GB of RAM, Windows XP Operating System and C++ programming language.²Overall rate is obtained by combining correct recognition rates of training and validation datasets.

Table 5. Credit risk evaluation neural network's learning schemes and their implementation results after 50000 iterations.

Learning scheme	LS1	LS2	LS3	LS4	LS5	LS6	LS7
Training-to-validation ratio	100:590	200:490	300:390	345:345	390:300	400:290	500:190
Training-to-validation percent	14.5%–85.5%	29%–71%	43.5%–56.5%	50%–50%	56.5%–43.5%	58%–42%	72.5%–27.5%
Obtained error	0.010	0.018	—	0.013	0.009	0.021	0.018
Training time (seconds) ¹	109.44	146.91	—	194.81	232.77	211.61	244.25
Run time (seconds) ¹	2.92×10^{-4}	2.55×10^{-4}	—	2.26×10^{-4}	3.13×10^{-4}	3.76×10^{-4}	4.95×10^{-4}
Training dataset correct evaluation rate	(99/100) 99%	(196/200) 98%	—	(341/345) 98.84%	(386/390) 98.97%	(392/400) 98%	(490/500) 98%
Validation dataset correct evaluation rate	(273/590) 46.27%	(394/490) 80.41%	—	(274/345) 79.42%	(235/300) 78.33%	(231/290) 79.66%	(157/190) 82.63%
Overall evaluation rate ²	(372/690) 53.91%	(590/690) 85.51%	—	(615/690) 89.13%	(621/690) 90%	(623/690) 90.29%	(647/690) 93.77%

¹Using a 2.8 GHz PC with 2 GB of RAM, Windows XP Operating System and C++ programming language.²Overall rate is obtained by combining correct recognition rates of training and validation datasets.

iterations required to converge to this error value under any scheme is marked, and then the neural network is trained for the marked number of iterations under the remaining six schemes.

The neural network implementation following learning (LS3) required the least number of iterations (20801 iterations), to converge to the required error value of 0.003. In fact, LS3 was the only learning scheme under which the neural network converged to the required minimum error value within the maximum iteration limit of 50000 iterations.

Therefore, the neural network model was also trained under all seven schemes for 20801 iterations and the implementation results, including training and validation (one feed forward pass of the

trained neural network) time costs, as well as correct evaluation rates using training and validation data were recorded as shown in Table 4.

Under learning scheme LS3, the neural network learnt and converged after 20801 iterations and within 73.92 seconds, whereas the running time for the trained neural network, using one forward pass, i.e., the time taken by the trained neural network to evaluate a credit application, and decide whether to accept or reject, was 0.38×10^{-4} seconds. The neural network was not trained for 50000 iterations as it converged to the required error value with less iteration (see Table 5).

The correct evaluation rate of the proposed neural system is of utmost importance. Tables 4 and 5

show the correct evaluation rates under all seven schemes. There are two evaluation rates: using the training dataset, and using the validation dataset. Since the neural network is trained over and over using the training dataset, we believe that the evaluation rates using the validation dataset are more significant, and show whether the neural network acquired meaningful learning or not. Following scheme LS3, the neural network model yielded 100% correct evaluation rate when using the training dataset as would be expected, whereas using the validation dataset yielded 81.03% correct evaluation rate. Combining results using validation and training datasets yields an overall correct evaluation rate of 89.28% for LS3.

The obtained result when using the training dataset does not mean that the trained neural network has memorized the examples (i.e., over-trained); as the successful training of the network is reflected by the obtained results when using the validation dataset which was not presented to the network previously. The problem of overtraining can generally occur when using supervised neural networks; and a solution to this problem was suggested in Ref. 47 which proposed a regularization neural network that has an error function consisting of a standard error term and a regularization term.

Careful inspection of the results in Tables 4 and 5 shows other learning schemes with higher overall evaluation rates. When training the neural network for 20801 iterations, scheme LS7 (500:190) appears to have an overall correct evaluation rate of 92.9%. Also when training the network for 50000 iterations, scheme LS7 has the highest overall evaluation ratio of 93.77%. However, these ratios can be misleading, because the neural network was exposed during training to a very large number of cases (500 cases), in comparison to the number of validation cases (190). In summary, we consider that the decision, upon which learning scheme is ideal, depends on:

- Achieving the required convergence error for higher classification accuracy.
- Requiring the least iterations for maintaining lower computational and time costs.
- Completion of training in a short time.
- Quick decision making once training is completed.
- Achieving the highest correct validation dataset ratio, i.e., data not exposed to the network before.

Based on the above considerations, we conclude that the credit risk evaluation neural network is most efficient when trained under the LS3 learning scheme.

Further work has also been carried on to investigate the performance of the proposed system using a single-hidden layer neural network (SHNN), when compared to using a double-hidden layer neural network (DHNN). We suggested in the first section that using a SHNN would reduce the computational cost.

Under learning scheme LS3 (300:390), which we propose to be the ideal training scheme based on the above investigation, we train and test a DHNN which has the same parameters as the SHNN, but with an extra hidden layer that has a single neuron. The number of neurons in this extra hidden layer is kept minimal in order to maintain the computational costs as low as possible despite the addition of this extra hidden layer. The DHNN was trained for a similar number of iterations as the SHNN; namely 20801 iterations, so that a more objective comparison can be made. Figures 2 and 3 show respectively

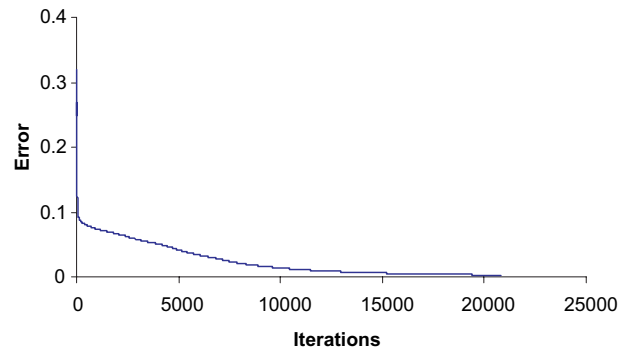


Fig. 2. Learning curve of the SHNN model under learning scheme LS3.

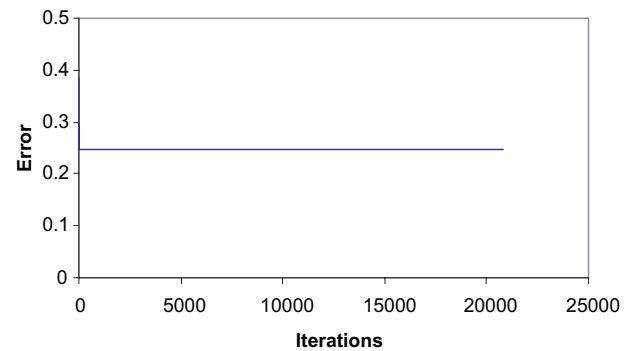


Fig. 3. Learning curve of the DHNN model under learning scheme LS3.

Table 6. Performance comparison between the SHNN and the DHNN models for credit risk evaluation.

Neural network	SHNN	DHNN
Input layer nodes	14	14
Hidden layer 1 nodes	9	1
Hidden layer 2 nodes	—	9
Output layer nodes	1	1
Learning coefficient	0.00935	0.00935
Momentum rate	0.67	0.67
Random initial weight range	-0.30+0.30	-0.30+0.30
Required minimum error	0.003	0.003
Obtained error	0.003	0.247
Iterations	20801	20801
Training time (seconds) ¹	73.92	85.53
Run time (seconds) ¹	0.38×10^{-4}	1.62×10^{-4}
Training dataset correct	(300/300)	(167/300)
evaluation rate	100%	55.67%
Validation dataset correct	(316/390)	(240/390)
evaluation rate	81.03%	61.54%
Overall evaluation rate ²	(616/690)	(407/690)
	89.28%	58.99%

¹Using a 2.8 GHz PC with 2 GB of RAM, Windows XP Operating System and C++ programming language.

²Overall rate is obtained by combining correct recognition rates of training and validation datasets.

the learning curves of the SHNN and the DHNN systems, whereas Table 6 describes the final parameters of both neural models as well as the evaluation rates.

Careful inspection of Table 6 shows that the use of a single-hidden layer neural network (SHNN) yields higher correct evaluation rates, and a reduced time cost when compared to using a double-hidden layer neural network (DHNN) for credit risk evaluation. Both training and run (one forward pass) times are higher with the DHNN, moreover the obtained error value for DHNN after the same number of iterations as the SHNN, is very high. In conclusion, using a SHNN for the task of credit risk evaluation, provides more superior results than when using a DHNN, which is one of the problems we aimed at investigating in this work.

5. Conclusions

In this paper, a novel approach to credit risk evaluation using a neural network was presented. In our approach we train a three-layer supervised neural network, which is based on the back propagation learning algorithm, following seven 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 a 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 Sec. 1. We also described in detail (Sec. 4) the criteria and considerations which are to be made in order to decide upon an ideal learning scheme. 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 a single-hidden layer neural network (SHNN) model; many previous works generally use double-hidden layer (DHNN) models for similar applications.

In order to implement our proposed neural system for credit risk evaluation, we used the Australian credit data. This is a real world dataset that has been successfully used for credit scoring and evaluation systems in previous works. The Australian credit data consists of 690 cases or instances: 307 instances of creditworthy applicants and 383 instances where credit is not creditworthy.

In conclusion, the credit risk evaluation neural network model performs best when using the LS3 learning scheme, which uses 300 cases for training and 390 cases for validation (43.5%:56.5%). We consider this learning scheme as ideal following the criteria which is described in Sec. 4, such as achieving the required error of 0.003, speed of training, speed of one-forward pass of the trained neural model, and the correct evaluation rates when using training and validation datasets. Under learning scheme LS3, correct evaluation rates of 100% and 81.03% were obtained using the training and validation data sets, respectively. Training the neural model was completed in 73.92 seconds, whereas the decision making time for the trained neural model was a fast 0.38×10^{-4} seconds.

We also compared the performance of the SHNN and the DHNN under the chosen ideal learning scheme LS3. Our investigation suggests that the SHNN outperforms the DHNN when applied to credit risk evaluation. Therefore, we suggest that this neural system based on a SHNN model can

be efficiently used in automatic processing of credit applications.

Future work will include comparing the presented neural network model when used for credit risk evaluation with other emerging neural models. Furthermore, further 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

1. K. K. Lai, L. Yu, S. Y. Wang and L. G. Zhou, Credit risk analysis using a reliability-based neural network ensemble model, *Lecture Notes in Computer Science* **4132** (2006) 682–690.
2. S. T. Li, W. Shiue and M. H. Huang, The evaluation of consumer loans using support vector machines, *Expert Systems with Applications* **30** (2006) 772–782.
3. A. Laha, Building contextual classifiers by integrating fuzzy rule based classification technique and *k*-nn method for credit scoring, *Advanced Engineering Informatics* **21** (2007) 281–291.
4. A. Vellido, P. J. G. Lisboa and J. Vaughan, Neural networks in business: A survey of applications (1992–1998), *Expert Systems with Applications* **17**(1) (1999) 51–70.
5. B. Panakkat and H. Adeli, Neural network models for earthquake magnitude prediction using multiple seismicity indicators, *International Journal of Neural Systems* **17**(1) (2007) 13–33.
6. M. A. Kramer, F. L. Chang, M. E. Cohen, D. Hudson and A. J. Szeri, Synchronization measures of the scalp EEG can discriminate healthy from Alzheimers subjects, *International Journal of Neural Systems* **17**(2) (2007) 61–69.
7. A. Terai and M. Nakagawa, A neural network model of metaphor understanding with dynamic interaction based on a statistical language analysis, *International Journal of Neural Systems* **17**(4) (2007) 265–274.
8. P. Huang and Y. Xu, SVM-based learning control of space robotic capturing operation, *International Journal of Neural Systems* **17**(6) (2007) 467–477.
9. H. C. Lian and B. L. Lu, Multi-view gender classification using multi-resolution local binary patterns and support vector machines, *International Journal of Neural Systems* **17**(6) (2007) 479–487.
10. N. C. Schneider and D. Graupe, A modified LAM-STAR neural network and its applications, *International Journal of Neural Systems* **18**(4) (2008) 331–337.
11. B. Cyganek, Colour image segmentation with support vector machines: Applications to road signs detection, *International Journal of Neural Systems* **18**(4) (2008) 339–345.
12. C. Fyfe, W. Barbakh, W. C. Ooi and H. Ko, Topological mappings of video and audio data, *International Journal of Neural Systems* **18**(6) (2008) 481–489.
13. P. Gopych, Biologically plausible BSDT recognition of complex images: The case of human faces, *International Journal of Neural Systems* **18**(6) (2008) 527–545.
14. S. Haykin, *Neural Networks: A Comprehensive Foundation* (2nd ed.) (Addison-Wesley/Longman, Singapore, 1999).
15. K. A. Smith, *Introduction to Neural Networks and Data Mining for Business Applications* (Eruditions Publishing, Australia, 1999).
16. B. S. Ahn, S. S. Cho and C. Y. Kim, The integrated methodology of rough set theory and artificial neural network for business failure prediction, *Expert Systems with Applications* **18**(2) (2000) 65–74.
17. T. S. Lee, C. C. Chiu, C. J. Lu and I. F. Chen, Credit scoring using the hybrid neural discriminant technique, *Expert Systems with Applications* **23**(3) (2002) 245–254.
18. I. Becerra-Fernandez, S. H. Zanakakis and S. Walczak, Knowledge discovery techniques for predicting country investment risk, *Computers and Industrial Engineering* **43**(4) (2002) 787–800.
19. R. Malhotra and D. K. Malhotra, Evaluating consumer loans using neural networks, *Omega* **31**(2) (2003) 83–96.
20. B. Baesens, R. Sentiono, C. Mues and J. Vanthienen, Using neural network rule extraction and decision tables for credit-risk evaluation, *Management Science* **49**(3) (2003) 312–329.
21. Z. Huang, H. Chen, C. J. Hsu, W. H. Chen and S. Wu, Credit rating analysis with support vector machines and neural networks: A market comparative study, *Decision Support Systems* **37**(4) (2004) 543–558.
22. T. S. Lee and I. F. Chen, A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines, *Expert Systems with Applications* **28**(4) (2005) 743–752.
23. D. West, S. Dellana and J. Qian, Neural network ensemble strategies for financial decision applications, *Computers and Operations Research* **32**(10) (2005) 2543–2559.
24. N. C. Hsieh, Hybrid mining approach in design of credit scoring model, *Expert Systems with Applications* **28** (2005) 655–665.
25. W. Huang, Y. Nakamori and S. Y. Wang, Forecasting stock market movement direction with support vector machine, *Computers and Operations Research* **32**(10) (2005) 2513–2522.
26. J. H. Min and Y. C. Lee, Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters, *Expert Systems with Applications* **28**(4) (2005) 603–614.
27. B. Baesens, T. Van Gestel, M. Stepanova, D. Van den Poel and J. Vanthienen, Neural network survival

- analysis for personal loan data, *Journal of the Operational Research Society* **56**(9) (2005) 1089–1098.
28. J. H. Min and Y. C. Lee, A practical approach to credit scoring, *Expert Systems with Applications* **35** (2008) 1762–1770.
 29. M. K. Lim and S. Y. Sohn, Cluster-based dynamic scoring model, *Expert Systems with Applications* **32** (2007) 427–431.
 30. D. Martens, B. Baesens, T. Van Gestel and J. Vanthienen, Comprehensible credit scoring models using rule extraction from support vector machines, *European Journal of Operational Research* **183** (2007) 1466–1476.
 31. C. L. Huang, M. C. Chen and C. J. Wang, Credit scoring with a data mining approach based on support vector machines, *Expert Systems with Applications* **33** (2007) 847–856.
 32. H. Abdou, J. Pointon and A. El-Masry, Neural nets versus conventional techniques in credit scoring in Egyptian banking, *Expert Systems with Applications* **35** (2008) 1275–1292.
 33. E. Angelini, G. di Tollo and A. Roli, A neural network approach for credit risk evaluation, *The Quarterly Review of Economics and Finance* **48** (2008) 733–755.
 34. L. Yu, S. Y. Wang and K. K. Lai, Credit risk assessment with a multistage neural network ensemble learning approach, *Expert Systems with Applications* **34** (2008) 1434–1444.
 35. C. F. Tsai and J. W. Wu, Using neural network ensembles for bankruptcy prediction and credit scoring, *Expert Systems with Applications* **34** (2008) 2639–2649.
 36. R. Setiono, B. Baesens and C. Mues, Recursive neural network rule extraction for data with mixed attributes, *IEEE Transactions on Neural Networks* **19**(2) (2008) 299–307.
 37. M. Šušteršič, D. Mramor and J. Zupan, Consumer credit scoring models with limited data, *Expert Systems with Applications* **36**(3) (2009) 4736–4744.
 38. A. Asuncion and D. J. Newman, UCI Machine Learning Repository [<http://www.ics.uci.edu/~mlearn/MLRepository.html>]. Irvine, CA: University of California, School of Information and Computer Science (2007).
 39. E. Boros, P. L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz and I. Muchnik, An implementation of logical analysis of data, *IEEE Transactions on Knowledge and Data Engineering* **12**(2) (March/April 2000) 292–306.
 40. B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens and J. Vanthienen, Benchmarking state-of-art classification algorithms for credit scoring, *Journal of the Operational Research Society* **54** (2003) 627–635.
 41. Y. S. Kim and S. Y. Sohn, Managing loan customers using misclassification patterns of credit scoring model, *Expert Systems with Applications* **26** (2004) 567–573.
 42. A. F. Atiya, Bankruptcy prediction for credit risk using neural networks: A survey and new results, *IEEE Transactions on Neural Networks* **12**(4) (2001) 929–935.
 43. S. Sakprasat and M. C. Sinclair, Classification rule mining for automatic credit approval using genetic programming, *Proceedings of the IEEE Congress on Evolutionary Computation* (Singapore, September 2007), pp. 548–555.
 44. A. Khashman, Blood cell identification using a simple neural network, *International Journal of Neural Systems* **18**(5) (2008) 453–458.
 45. A. Khashman and B. Sekeroglu, Document image binarisation using a supervised neural network, *International Journal of Neural Systems* **18**(5) (2008) 405–418.
 46. B. Hamers, J. A. K. Suykens and B. De Moor, Coupled transductive ensemble learning of kernel models, *Journal of Machine Learning Research* **1** (2003) 1–48.
 47. H. Adeli and M. Wu, Regularization neural network for construction cost estimation, *Journal of Construction Engineering and Management* **124**(1) (1998) 18–24.

Copyright of International Journal of Neural Systems is the property of World Scientific Publishing Company and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.