Option 1: Model development and report

Introduction

This report explores the predictive capability of neural networks on a binary classification problem. Using the credit-screening database, the idea is to train multiple neural networks and compare the predictive power in either accepting or rejecting a credit application. In the context of credit screening, there is a conservative preference towards rejecting bad risks rather than overall goodness of fit. While this work uses the percentage of correctly classified as the performance metric, it can be easily extended to include the false negative error as part of the model loss. In combination with various optimisers, this study compares the results of Adaptive moment estimation (Adam) and Stochastic Gradient Descent (SGD). Section 1 will describe the initial data processing steps and exploratory data analysis then section 2 contains the modelling information, results and discussion.

Section 1: Data processing

1. The data consisted of 689 observations, with 15 feature variables and 1 target variable. Initial inspections found 67 instances of missing values, denoted by '?'. Removing the rows containing missing values reduced the dataset to 652 observations, as some rows contained multiple missing values.
   The next step converted all feature variables to numeric values. The feature variables will then be normalised via z-score and binarize the target variable in preparation for modelling.

Variable	Initial data type	Data processing method
b	Character with 2 levels	Integer encoding
30.83	Character of decimals	As numeric
0	Character of decimals	As numeric
u	Character with 3 levels	Integer encoding
g	Character with 3 levels	Integer encoding
w	Character with 14 levels	Integer encoding
v	Character with 9 levels	Integer encoding
1.25	Decimals	N/A
t	Boolean	N/A
t_1	Boolean	N/A
01	Character of integers	As numeric
f	Boolean	N/A
g_1	Character with 3 levels	Integer encoding
00202	Character of integers	As numeric
0_1	Integers	N/A
+	Character with 2 levels	Integer encoding

Table 1: Variable data types

2. Normalising the variables shifts features to a similar range, such that the model weights them similarly. Doing this also allows simple visualisation of the shape of the data. The numerical variables are displayed via boxplot in Figure 1a, and it is noted that most of these feature variables are positively skewed. This is expected as many of these variables take nonnegative values with a large mass at 0. Also, variable '0_1' is especially small due to the high concentration of values at 0 and by the nature of the boxplot, many values were considered as outliers, however, are not shown on the graph.
   Meanwhile, the categorical variables are displayed via frequency polygon in Figure 1b. As many of these variables are binary, there exist two peaks in these cases and are shifted such that the mean of the two values is 0. For example, variable 'g_1' has a large peak near 0 and a smaller peak around 3 and due to the distribution of values, the overall mean is still 0. Similar arguments can be made for the other binary variables.

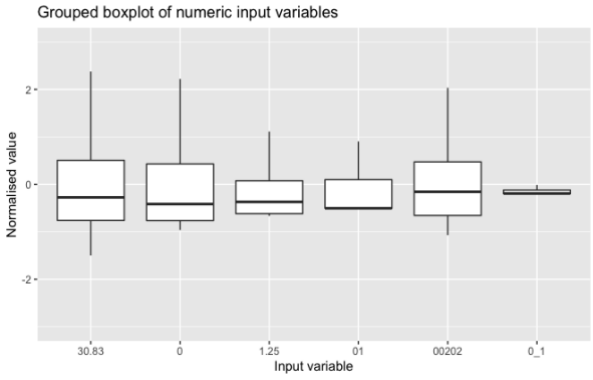

Figure 1a: Boxplot of normalised numeric variables

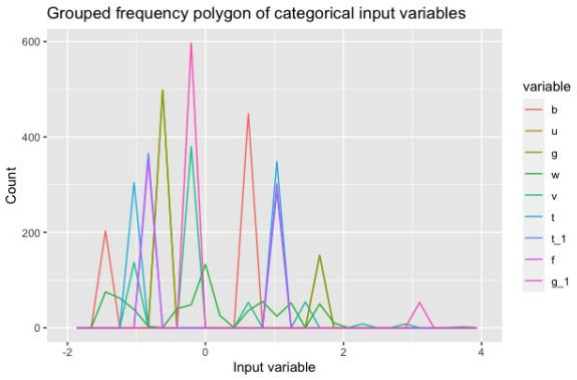

Figure 1b: Frequency polygon of normalised categorical variables

3. A correlation heatmap of the features is also included. This gives visualisation of the interactions between input variables and identifies any potential redundancies which can be excluded from the model. Figure 2 shows that most variables are weakly correlated, with variables 'u' and 'g' appearing perfectly correlated. Performing an element-wise comparison of 'u' and 'g' confirms that these two variables provide identical information, hence one variable may be removed. In this case, variable 'g' is removed from the model dataset.

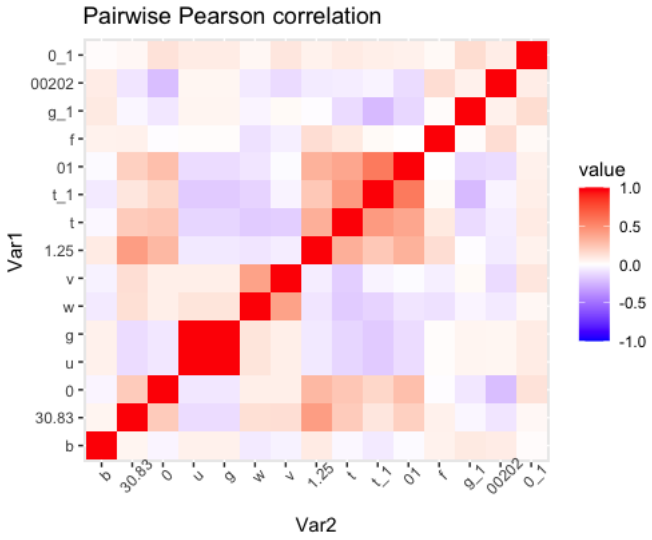

Figure 2: Correlation heatmap of features

4. Additional exploratory data analysis is also performed. It is important to test the distribution of the target variable for any imbalances. The reason for this is if there an imbalance of the target classes, the model could always pick the majority class irrespective of the input features. For example, if 95% of the data is TRUE and 5% is FALSE, the model achieves 95% correct classification by trivially guessing TRUE for all cases. This means only 5% are misclassified and is known as the null error rate, thus the normal accuracy measure will not provide meaningful information.
   A quick summary reveals that 45% of the target class is TRUE and 55% are FALSE. This means that the null error rate will be 45% and the model should aim to perform better than this.

Section 2: Model

The model used in this study was a neural network with a single hidden layer. Different optimisers were chosen and compared, as well as regularisation techniques. The activation function for the hidden layer was chosen as the Rectified Linear Unit (ReLU), which is preferred in modern neural networks due to ease of computation and storage. The output layer used the default softmax activation function from the ANN2 package. Table 2 describes the chosen parameters.

	SGD	Adam
Learning rate	0.01	0.01
Hidden layer nodes	14	14
Epochs	150	100

Table 2: Model parameters

Preliminary trial experiments revealed that a learning rate of 0.01 performed well for both SGD and Adam. The number of nodes in the single hidden layer was chosen to be equal to the number of input features (14 after excluding the perfectly correlated 'g') as a moderate amount to allow for sufficient degrees of freedom in calculations. The number of epochs for Adam was chosen as 100 due to the fast convergence of the error to 0 and the number of epochs for SGD was chosen as 150 due to the bootstrap nature of SGD sampling, since approximately 1/3 of the data isn't selected per epoch.

1. Using the percentage correctly classified metric, Table 3 shows the results for the base case of SGD and Adam. The mean is taken over 10 trial experiments and the standard deviation is reported in parentheses. The results for SGD and Adam are quite similar, though SGD required more training epochs to converge. If SGD had used the same number of epochs as Adam, it is likely that Adam would show much greater accuracy due to the irregular nature of SGD. Both SGD and Adam reveal a test accuracy just over 80% and in both cases, the mean test accuracy is lower and the standard deviation is higher, which is expected since the model is optimised for the training set. In relation to Section 1.4, the test errors (approximately 16-18%) is significantly lower than the null error rate of 45%, meaning that these models perform much better than a trival guess.

Percent correctly classified	SGD	Adam
Training	99.79% (0.29%)	99.75% (0.48%)
Test	84.48% (1.26%)	82.94% (1.02%)

Table 3: Percent correctly classified for SGD and Adam

2. [Option II] Next, the effect of the learning rate is explored on SGD. Using the same variables as previous but ranging the learning rate parameter from 0.0001 to 0.1, Figure 3 shows the training and test performance for the learning rates on a log scale. The decision for the optimal learning rate should be chosen from the highest test accuracy, which is at 0.005. Note that there is not a significant difference in the accuracy for neighbouring values of the learning rate, however it is quite clear that the performance drops off at the extreme values, hence a middle point such as 0.005 would be suitable.

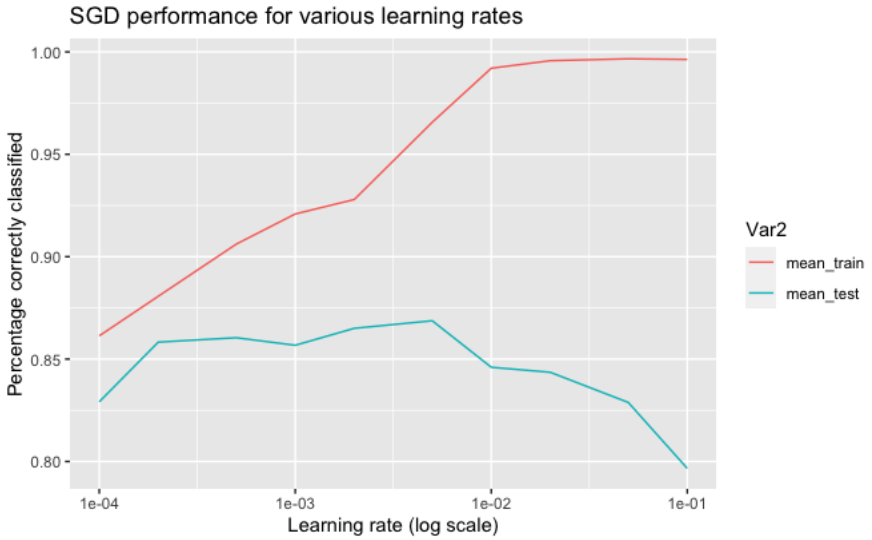

Figure 3: SGD performance for various learning rates

3. [Option II] Using the selected 0.005 as the learning rate, the detailed performance measures for this model are displayed. Table 4 shows the confusion matrix for this model on the test set (since training accuracy is not important). Looking at the errors in the model, there are 32 false negatives and 20 false positives. While these numbers are small relative to the correct predictions, the application context plays a role here, as credit-screening should prioritise minimising false negatives, due to high possibility of default. Potential solutions will be detailed next.

Prediction \ Reference	FALSE	TRUE
FALSE	153	32
TRUE	20	121

Table 4: Confusion matrix SGD with learning rate 0.005

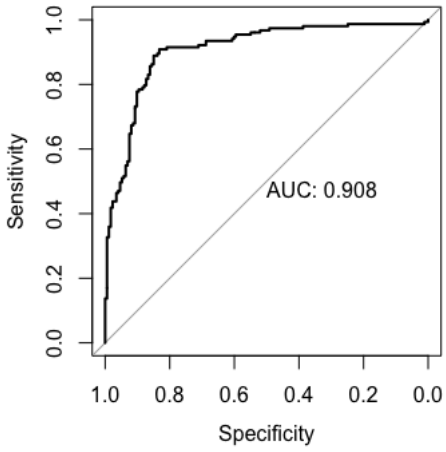

Figure 4: ROC with printed AUC for SGD with learning rate 0.005

4. There are a few limitations with the model and the methodology.
   Firstly, many of the categorical variables were encoded as integers to avoid dimensionality inflation from methods such as one-hot encoding. However, it is not known whether any of these variables are ordinal and if so, what the ranking of the classes are, especially since most feature variables were categorical. While other models such as tree-based methods would handle categorical data with ease, this is one of the limitations in using the neural network. Therefore, a solution to this would be to fit other models more robust to categorical variables.
   Next, while the neural network model is highly flexible, it is difficult to understand the impact of each predictor due to the net of interactions between nodes. Continuing from the first point, the lack of understanding of what the feature variables mean also reduces interpretability of the model and learning more about these predictors may allow improvements to the model.
   Further, the contextual issue of credit-screening should prioritise the detection of bad risks, that is, minimise false negatives. Since the percentage of correctly classified measure is used here, it is found that there is a higher number of false negatives than false positives. There are a few methods to overcome this. One way would be to decrease the threshold for marking an individual as a bad risk, which would decrease the false negative rate and increase the false positive rate, such that the credit supplier will service a higher proportion of true good risks whilst rejecting the rest. An alternative method would be to impose higher weights on the observations in the data who are bad risks such that the model prioritises this detection.
   Then finally, the modelling process used in this study was mainly for comparison purposes between SGD and Adam under similar conditions. This means that these models have not been optimised and should not be used as actual predictions for credit screening. Future research could implement regularisation for these models and compare the results for a large number of epochs to measure the impact of overfitting.

Conclusion

Overall, this study managed to demonstrate the predictive capabilities of neural networks with SGD and Adam optimisers. While the models were not optimised and may have had issues with the categorical variables, both SGD and Adam yielded a significantly higher accuracy than a trivial guess. SGD and Adam had similar training and test results when the number of epochs was 150 for SGD and 100 for Adam, however reducing the number of epochs such that both methods used the same parameters would show better results from Adam.

References

Friedman, J. H. (2017). The elements of statistical learning: Data mining, inference, and prediction. springer open.

ggplot2 : Quick correlation matrix heatmap - R software and data visualization - Easy Guides - Wiki - STHDA. (2021). Retrieved 25 September 2021, from http://www.sthda.com/english/wiki/ggplot2-quick-correlation- matrix-heatmap-r-software-and-data-visualization

Hassanat, A. (2021). Data normalization and standardization in neural networks. Retrieved 16 October 2021, from https://stats.stackexchange.com/questions/7757/data-normalization-and-standardization-in-neural-networks

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

Navlani, A. (2021). Neural Network Models in R. Retrieved 16 October 2021, from https://www.datacamp.com/community/tutorials/neural-network-models-r