Modelling Credit Risk with Neural Networks

Darrius Lam

This report will aim to explore the use of neural networks in modelling a credit risk problem. In particular, the neural networks discussed will be a Dense Feed Forward Neural Network and a Wide and Deep Neural Network.

1. Problem Specification

The specified problem is to classify a person as a good or bad credit risk based on a set of attributes. This is a binary classification problem. The chosen metrics will be accuracy, precision, recall and f-score.

2. Dataset and Preprocessing

The dataset consists of German credit data sourced from UC Irvine (1994). There are 1000 instances and 20 features (See Appendix 1).

The output class has been defined as ‘0’ being the bad credit risk class and ‘1’ being the good credit risk class.

Variables ‘checking_status’, ‘savings_status’, ‘employment’, ‘class’, ‘own_telephone’ and ‘foreign_worker’ were ordinally encoded.

Unordered categorical variables ‘credit_history', 'purpose', 'personal_status', 'other_parties', 'property_magnitude', 'other_payment_plans', 'housing', 'job', 'own_telephone', 'foreign_worker', 'class' were one-hot encoded.

The remaining variables 'duration', 'credit_amount', 'age', ‘residence_since’ and ‘existing_credits’ were untouched as they were in appropriate formats. Continuous features ‘duration’, ‘credit_amount’ and ‘age’ were standard scaled after splitting into training and test sets.

Datasets were split into training, validation and testing sets. Considering the small size of the dataset, splits of 80% training, 10% validation and 10% validation were chosen. After experimentation, this split appeared to have performed the best on the neural networks chosen.

3. Exploratory Data Analysis

Through data analysis, it appears that the most striking variables are, credit_amount, age and duration, with correlation values of -0.83, 0.77 and -0.93 respectively. Means are 3271.26, 35.55 and 20.93 respectively and all 3 variables are right skewed (Figure 1). The parameters relevant to these features will be crucial in the predictive modelling and may be emphasised through the Wide and Deep neural network architecture.

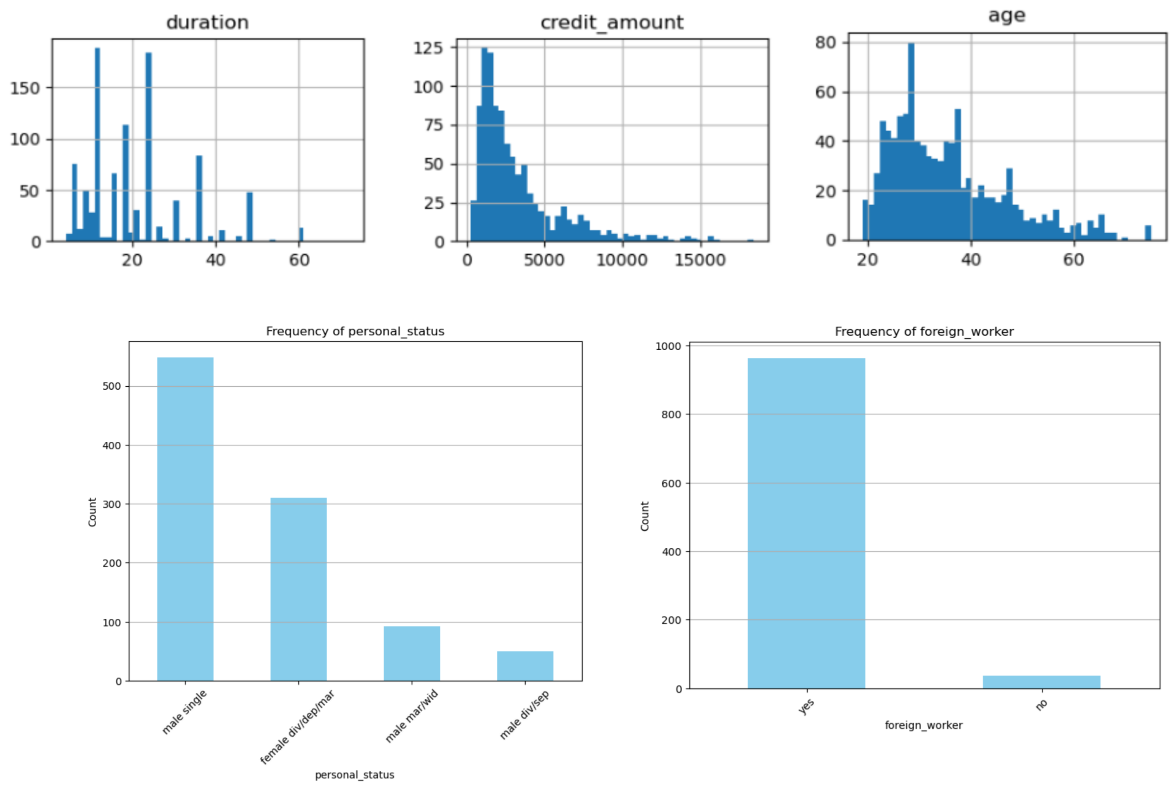


Figure 1. Plots of frequency of features

There is some concern for data set bias due to the more linear distribution of age, which may not be representative for debtees. Furthermore, there is a large overrepresentation of male debtees compared to the population which may reflect bias (Figure 1). One striking variable is that foreign workers dominate the dataset at a ratio of 96:4 and this may mean a deficiency in the resulting model in more evenly distributed datasets (Figure 1). The distributions of these three features may hinder the predictive success of the resulting model on more balanced datasets.

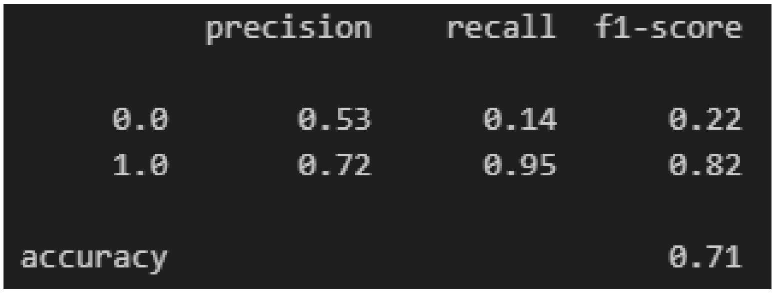
Additionally there is a large imbalance of the output classification of 7:3, good credit risk to bad credit risk (Figure 1), which may hinder a neural network's predictions. This may require class weighting in the fitting of the model, although in testing it appeared detrimental to the relevant metrics.

4. Benchmark Model

Given the high negative correlation of the 'duration' feature with a value of p = -0.93, a linear regression model using this variable would be suitable. This would provide a simple, intuitive and competitive benchmark model.
After fitting the model with my data, a linear regression function is produced:

$$y = 0.87 + \ - 0.008 \cdot x$$

The coefficient is small due to a difference in scale between a continuous and binary variable. This model appears to have performed well, with a high accuracy of 0.71 and f-score of 0.82 for the good credit risk class (1.0). However upon further analysis, it appears that the model indiscriminately produces a ‘good credit risk’ classification, indicated by the polarised recall metric. This along with the fact that a similar 70% of the dataset consists of ‘good credit risk’, it can be concluded that the benchmark model performs poorly.



```
           precision    recall  f1-score

     0.0       0.53      0.14      0.22
     1.0       0.72      0.95      0.82

accuracy                           0.71
```

Figure 2. Benchmark Model Results

5. Two Chosen Neural Network Architectures

To solve this problem, two neural network architectures were chosen: a Dense Feed-Forward Network and Wide and Deep network.

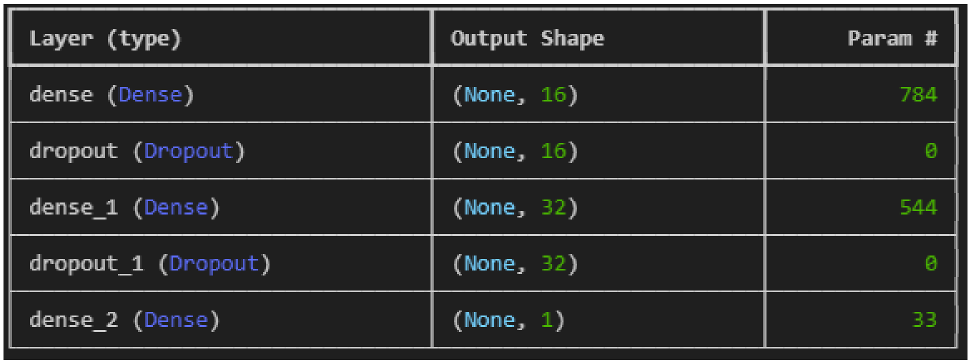
5.1 Dense Feed-Forward Network

A Dense Feed-Forward Neural Network is the first neural network architecture chosen for this problem. It consists of a number of layers of neurons, each of which are fully connected with the next layer. It mimics a human brain, in the sense that every connection creates a parameter which will be adjusted and the sum of which forms a complex relationship between features and the output.

It was chosen for its simplicity which allows it to serve as a baseline model compared to a more complex architecture.

Due to the nature of the data, dropout layers were included between each layer to reduce overfitting. After manual hyper-parameter tuning, a dropout rate of 0.5 was chosen to maximise test accuracy.

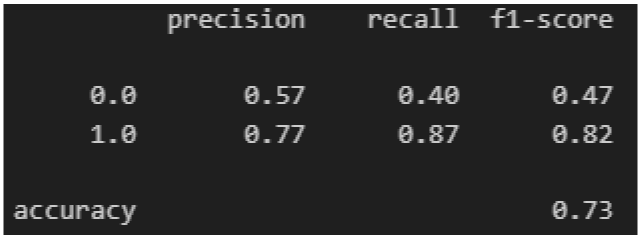
Automatic hyper-parameter tuning was also performed on: number of layers, neurons per layer and activation function, using keras-tuner. The final architecture consists of 2 hidden layers with 16 and 32 neurons in each layer, using the leaky-relu activation function (Figure 3).



Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	784
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 32)	544
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 1)	33

Figure 3. Dense Feed-Forward Neural Network Architecture

It produced results:



```
             precision    recall  f1-score

         0.0       0.57      0.40      0.47
         1.0       0.77      0.87      0.82

    accuracy                           0.73
```

Figure 4. Dense Feed Forward Neural Network Results

Compared to the benchmark model, accuracy has slightly improved two percentage points, with an identical f-score for the 'good credit risk' class. Recall and f-score improved significantly, from 22% to 47%, indicating that the model is correctly classifying some 'bad credit risk' data instances. This is a significant improvement from the benchmark model which is likely meritless.

5.2 Wide and Deep Neural Network

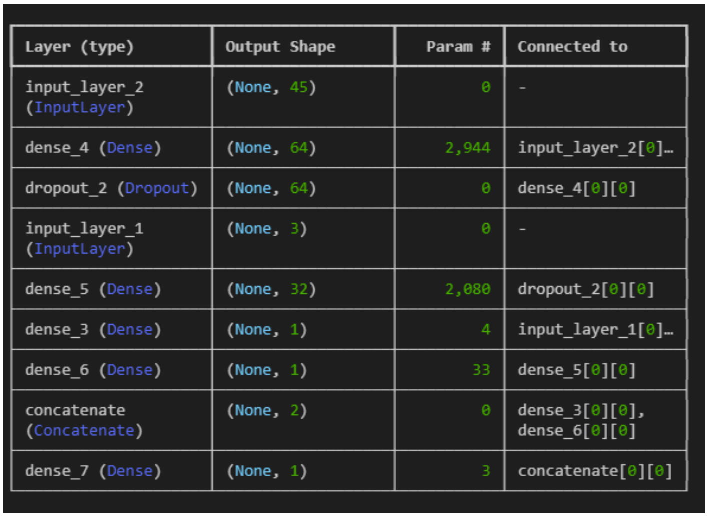
The second chosen neural network architecture is the Wide and Deep Neural Network. It consists of two sections, wide and deep. The wide section is essentially a linear model, and excels in modelling linear relationships. The deep section performs better in categorical elements as it consists of a few layers of dense neurons. This architecture was chosen because it combines the strengths of linear and deep learning, and fits the dataset as it contains both categorical and continuous elements. As this architecture uses Keras' Functional API it also allows for embedding as well as standard scaled features, which may improve the models ability to find relationships between categories.

The wide section consists of zero hidden layers and the deep section consists of 2 hidden layers. Through manual hyper-parameter tuning, this structure (Figure 6) was found to be the best.

Deep Layer Number	**Test Accuracy**	**Deep Activation Function**	**Test Accuracy**

2	0.71	Relu	0.71
3	0.68	Leaky-Relu	0.69
4	0.68	Tanh	0.67

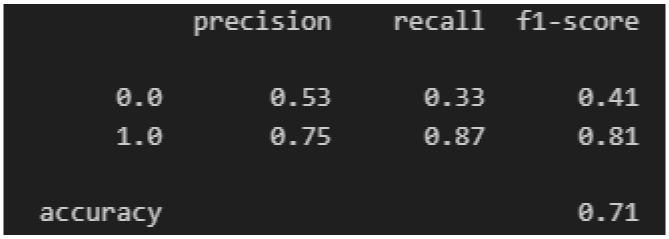
Figure 5. Hyper-parameter tuning. (Activation function was tested with 2 deep hidden layers)



```
Layer (type)           Output Shape     Param #   Connected to
input_layer_2          (None, 45)             0   -
(InputLayer)
dense_4 (Dense)        (None, 64)         2,944   input_layer_2[0]…
dropout_2 (Dropout)    (None, 64)             0   dense_4[0][0]
input_layer_1          (None, 3)              0   -
(InputLayer)
dense_5 (Dense)        (None, 32)         2,080   dropout_2[0][0]
dense_3 (Dense)        (None, 1)              4   input_layer_1[0]…
dense_6 (Dense)        (None, 1)             33   dense_5[0][0]
concatenate            (None, 2)              0   dense_3[0][0],
(Concatenate)                                     dense_6[0][0]
dense_7 (Dense)        (None, 1)              3   concatenate[0][0]
```

Figure 5. Wide and Deep Architecture

The results are:



```
              precision    recall  f1-score

         0.0       0.53      0.33      0.41
         1.0       0.75      0.87      0.81

    accuracy                           0.71
```

This model seems to perform almost identical to the Dense Feed Forward Network. This may be due to a lack of complexity and correlation in the categorical features, with the continuous features dominating the predictions.

6. Final Evaluation

Both models showed similar performance compared to each other. They both outperformed the benchmark model by a significant amount. Although the Wide and Deep Architecture is more complex and can better capture abstract relationships between categorical features, it is not evident in this dataset. This may be due to limitations on the size of the dataset or intrinsic limitations of the features.

References

Hofmann,Hans. (1994). Statlog (German Credit Data). UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data

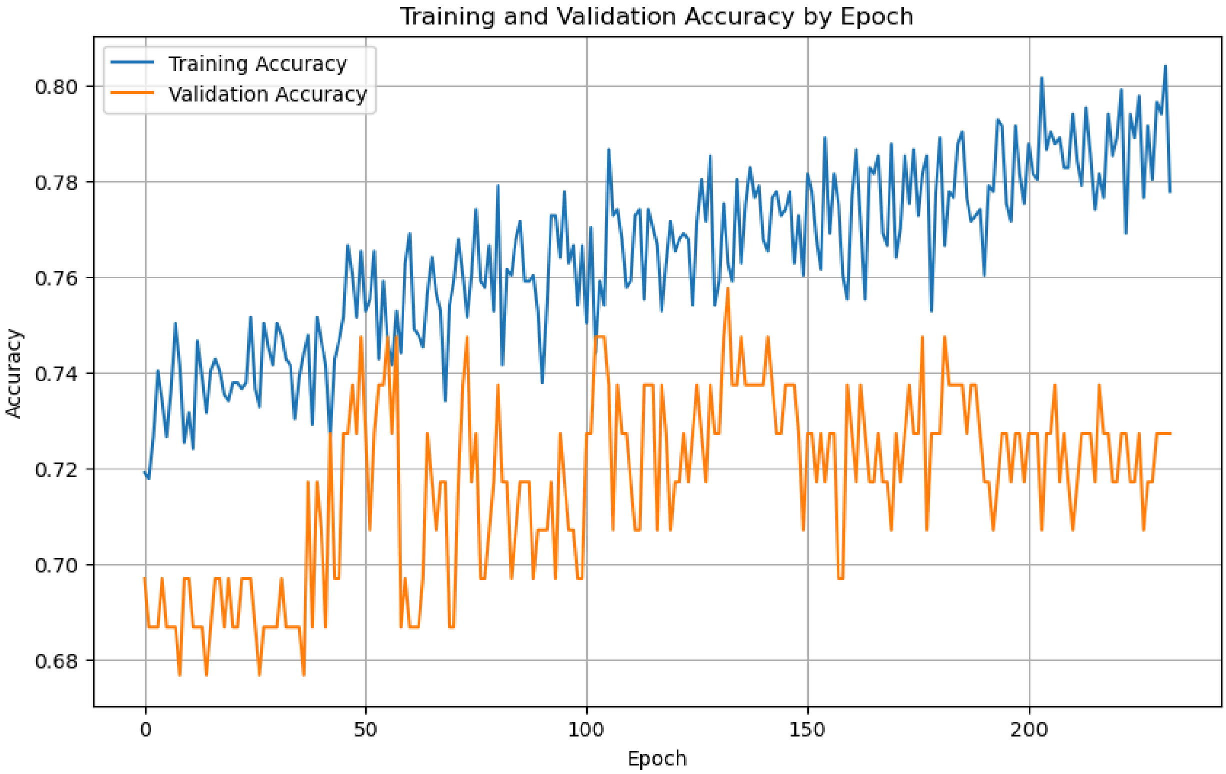
Appendix

1. Description of features and output class.

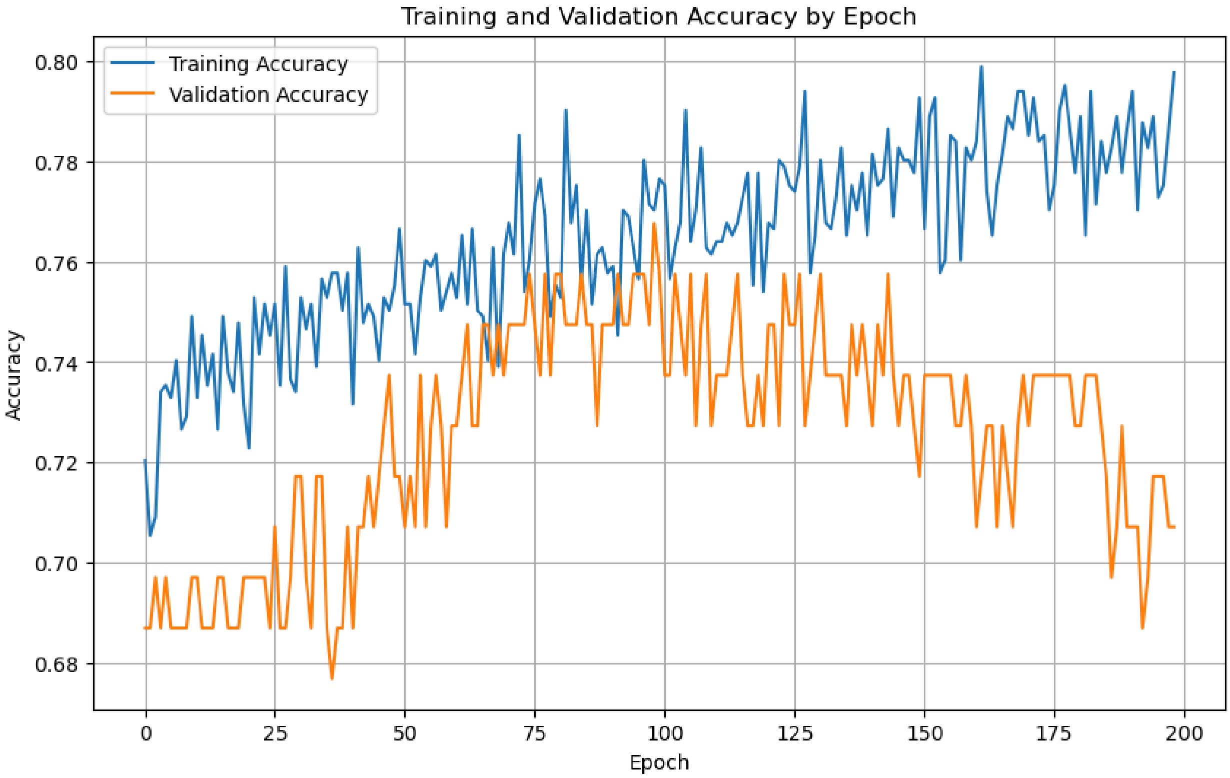
Variable Name	Description	Datatype
class	Credit risk classification (target variable) as good credit risk (value 1) and bad credit risk (value 0)	Nominal categorical
checking_status	Status of existing checking account (in Deutsche Mark)	Nominal categorical
duration	Duration with the company in months	Numeric
credit_history	Credit history	Nominal categorical
purpose	Purpose of the credit (e.g., car, television)	Nominal categorical
credit_amount	Credit amount	Numeric
savings_status	Status of savings account/bonds (in Deutsche Mark)	Nominal categorical
employment	Present employment in number of years	Nominal categorical
installment_commitment	Installment rate in percentage of disposable income	Numeric
personal_status	Personal status and sex	Nominal categorical
other_parties	Other debtors/guarantors	Nominal categorical
residence_since	Present residence since X years	Numeric

property_magnitude	Property (e.g., real estate)	Nominal categorical
age	Age in years	Numeric
other_payment_plans	Other installment plans (banks, stores)	Nominal categorical
housing	Housing status (rent, own)	Nominal categorical
existing_credits	Number of existing credits at this bank	Numeric
job	Job category	Nominal categorical
num_dependents	Number of people being liable to provide maintenance for	Numeric
own_telephone	Own telephone (yes, no)	Nominal categorical
foreign_worker	Foreign worker status (yes, no)	Nominal categorical

2. Dense Feed-Forward training to validation set accuracy



3. Wide and Deep training to validation set accuracy



Generative AI Usage

Generative AI was leveraged significantly in this project. It was used in general debugging, finding solutions to difficult problems and doing menial coding. Generative AI was also used largely in handling the datasets, due to unfamiliarity with Python syntax, data structure methods and general practices.

For example,

```
for col in data.columns:
    data[col] = data[col].apply(lambda x: x.decode('utf-8') if isinstance(x, bytes) else x)
```

this block was generated by AI to help convert byte strings to regular strings in the dataset and only then could it be processed normally.

Another example is:

```
plt.figure(figsize=(10, 6))
plt.plot(hist.history['loss'], label='Training Loss')
plt.plot(hist.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss by Epoch')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)
plt.show()
```

AI here helped create the graphs plotting Training Loss and Validation Loss.