Report Writeup – Neural Network(Reference Only)

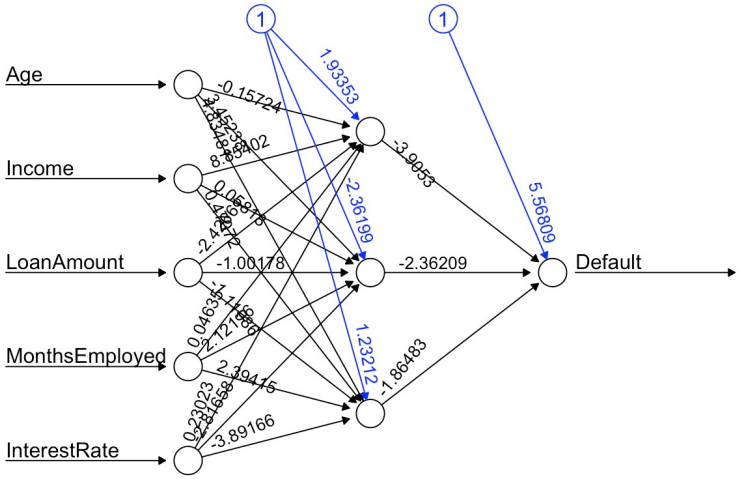
Karen

A neural network is a computational model inspired by the structure and functioning of the human brain. It consists of interconnected layers of nodes, or "neurons," which process data by passing it through weighted connections. Neural networks are widely used for tasks such as pattern recognition, classification, and prediction, as they are capable of learning complex relationships from data.

Hyperparameters are settings that define the structure and training of the neural network model, such as the number of layers, neurons, learning rate, and activation function. Choosing the right hyperparameters is essential for the model to learn patterns effectively and generalize well.

To optimize hyperparameters, cross-validations (CV) is used. It could splits the data into multiple training and validation sets, helping identify the settings that maximize performance metrics like accuracy or AUC. This process reduces overfitting and ensures the model's reliability.

The 5 variable I used are Age, Income, LoanAmount, MonthsEmployed, and InterestRate, based on the result of LASSO previously. Each of these variables connects to the neurons in the hidden layer through weighted connections. These weights indicate the strength and direction of the influence each input variable has on the respective hidden neurons.



The two most important variables I found out are Income and InterestRate. Positive weights, like Income's 8.83402, suggest that higher values increase the likelihood of avoiding default. Interstate, with negative weights such as -3.89166, shows that higher rates increase default risk. We can see that these weights reflect how variables differently influence the model's predictions.

A confusion matrix was also generated to assess the performance of the neural network model on the test data:

Predicted Class	Actual = 0	Actual = 1
Predicted = 0	4089	2000
Predicted = 1	1841	3930

The confusion matrix highlights a balanced performance in terms of distinguishing between default and non-default cases. The model successfully identifies 3930 true positives and 4089 true negatives, though it also misclassifies 3841(2000+1841) cases, which reflects the inherent trade-off in classification.

Then, we calculate the accuracy by this formula:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}}$$

We got the accuracy of 0.6761, which means that the model correctly predicts approximately 67.61% of cases in the test dataset, was an OK accuracy but not the best.