Neural Computing Group Coursework: Wine Quality

Initial Analysis and Work Done on Dataset:

The dataset from UCI is divided into red and white wine dataset, white wine dataset was used as it had more instances than the red wine dataset, it contained 4898 instances. It has 11 feature columns and 1 target multi-class column of wine quality ranging from the values 3-9, 3 being poor and 9 being excellent wine quality. The dataset does not contain any missing or null values and the distribution of the target class is heavily imbalanced where classes 5 and 6 heavily outnumbered the other classes, making up 75% of the population. The outliers in the dataset were removed using Interquartile range. The early versions of the model we trained on this dataset had a maximum accuracy of 55-60%, this may be due to the complex and imbalanced nature of the dataset. This problem was solved by feature engineering, as the dataset was split into 2 groups of qualities 3-5 which was labelled as 0 and 6-9 labelled as 1, making the work a binary classification task. Furthermore, the training features were normalised before being fed to the neural network. This improved the accuracy of the model and brought it to the range of 75-80%. The dataset was used in Cortez et al (2009), with NECO methods such as multi-layer perceptrons and support-vector machines (SVMs) being applied to it. In that paper, SVMs performed the best, although it was used for a regression task. This dataset was also used in Di (2022), where they rearranged the features using PCA and applied 1D-CNN and further optimised the features using backpropagation. In this paper, they also compared the performance of CNN to other methods and CNN gave the best accuracy of 83%.

Comparison of Python and Matlab:

PyTorch, pandas, numpy, matplotlib and time libraries was used in the implementation of the model in Python whereas in MATLAB, the neural net was trained using trainrp which is a network training function that updates weight and bias values according to the resilient backpropagation algorithm. Through experimentation by changing hidden layer sizes and learning rates, the parameters which led to the best performance of the model were found iteratively. It is intuitive that the higher the number of hidden neurons and epochs will lead to longer training time and this was true in this model's experimentation. During the experimentation, it was apparent that adjusting learning rates too drastically had an adverse effect on the model's performance. The final architecture of the model included 7 hidden-layer neurons, sigmoid function as activation function, 250 epochs and a learning rate of 0.4. The performance of the 2 implementations are compared below.

Python	Metric	Matlab
0.81 seconds	Training speed	18.9 seconds
79.12%	Test set accuracy	79%

Table 1: Python and Matlab performance comparison

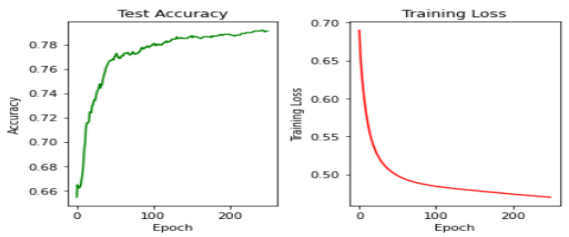

Figure 1: Python model performance

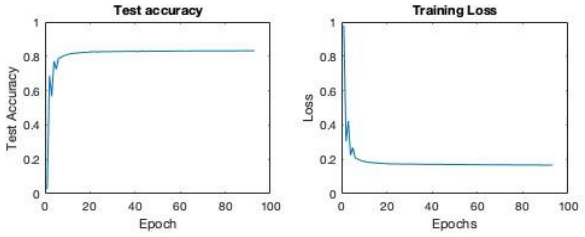

Figure 2: Matlab model performance

Summary: Python implementation was faster than Matlab and increasing the number of hidden neurons reduced the training speed while not improving the model accuracy very much after hitting a plateau. The value of learning rates also has a middle ground to improve the performance of the model and should be adjusted incrementally. For future work, the red and white wine dataset should be joined to add more instances and variety to the dataset used in the modelling, as well as using SVMs to perform multiclass-classification tasks.

References

Cortez P., Cerdeira A., Almeida F., Matos T., Reis J. (2009) 'Modelling wine preferences by data mining from physicochemical properties', *Decision Support Systems*, 47(4), pp. 547-553

Di, S. and Yang, Y. (2022) 'Prediction of red wine quality using one-dimensional convolutional neural networks', arXiv.org.