INM427 Neural Computing
Supanut Sookkho (MSc Data Science / 230024841) & Yumi Heo (Msc Data Science / 230003122)

A Comparison Between Python and MATLAB for Binary Classification in Predicting Bank Churn

1. Initial Analysis of the Data Set

The data set [1] from Kaggle has 14 features including a target and 165,034 samples.10 features remained after removing 3 features deemed unnecessary for classification. Normalisation was applied to the features for better accuracy. After checking the histogram of each feature and finding that some features and the target have unbalanced data, SMOTE was applied to ensure that the training set is balanced.

2. A Comparison Between Python and MATLAB in Model Implementation

A multilayer perceptron was implemented for this task. However, Support Vector Machines can be used for this task since it was originally made for binary classification [2]. ReLU activation was applied to the hidden layer nodes, and sigmoid to the output layer nodes. Grid search was implemented to find the best combination of the learning rate and the number of hidden neurons. The learning rate range is [0.01, 0.1, 0.3, 0.4] and the number of neurons is [10, 25, 50, 100, 200]. Although the number of hidden neurons should lie between the size of the input layer and the size of the output layer [3], there is another suggestion to use more than 100 neurons in the network [4]. Our model performed better with a larger number of neurons and a higher learning rate in our experiment (Table 1).

Programming Language	Python	MATLAB
Optimised Hyperparameters from Grid Search	Learning rate: 0.2 The number of neurons: 200	Learning rate: 0.3 The number of neurons: 25
Training Speed (Second)	1235.00	84.14
Test Set Accuracy of the Final Model	78.22%	85.53%
Precision, Recall, and F1 of the Final Model	48.78%, 69.30%, 57.26%	69.81%, 53.48%, 60.57%

Table 1

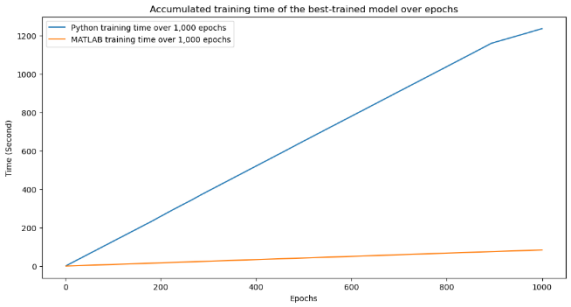

Figure 1

Figure 2

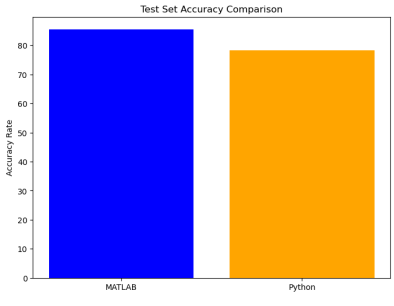

Figure 3

From this experiment, MATLAB is faster than Python when training the best model because MATLAB specializes in matrix computation [5] (Figure 1). The loss graph is one way to compare the training speed between MATLAB and Python as well. MATLAB started with a larger loss when training and validating in the first epoch (Figure 2). Then, MATLAB decreased the loss after passing 14 epochs whereas Python started with relatively low losses, and the magnitude of loss reduction is relatively small compared to MATLAB. We can say MATLAB converges faster than Python. The hypothesis behind this conclusion is that the initial weights in MATLAB are set higher than in Python in the first epoch. Despite the large weight setting, MATLAB's optimizer is likely to find the local minima faster than Python in non-convex scenarios. Additionally, differences in the randomness during training, such as weight initialization, may contribute to differences between Python and MATLAB. MATLAB's faster performance may be from its neural network with 25 neurons, compared to Python's 200, enabling quicker forward and backward propagations. Eventually, MATLAB achieves higher test accuracy than Python (Figure 3), influenced by differing hyperparameters, loss reduction rates during training, and variations in weight settings.

References

[1] W. Reade and A. Chow, 'Binary Classification with a Bank Churn Dataset'. Kaggle, Jan. 02, 2024. Accessed: Feb. 05, 2024. [Online]. Available: https://www.kaggle.com/competitions/playground-series-s4e1

[2] C. M. Bishop, *Pattern recognition and machine learning*. in Information science and statistics. New York: Springer, 2006.

[3] J. Heaton, 'The Number of Hidden Layers', Heaton Research. Accessed: Feb. 16, 2024. [Online]. Available: https://www.heatonresearch.com/2017/06/01/hidden-layers.html#:~:text=The%20number%20of%20hidden%20neurons%20should%20be%202%2F3%20the,size%20of%20the%20input%20layer.

[4] A. Koutsoukas, K. J. Monaghan, X. Li, and J. Huan, 'Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data', J Cheminform, vol. 9, no. 1, p. 42, Dec. 2017, doi: 10.1186/s13321-017-0226-y.

[5] J. Unpingco, 'Some Comparative Benchmarks for Linear Algebra Computations in MATLAB and Scientific Python', in 2008 DoD HPCMP Users Group Conference, Seattle, WA, USA: IEEE, 2008, pp. 503–505. doi: 10.1109/DoD.HPCMP.UGC.2008.49.