INM427: Neural Computing
Dimitris Nikandrou & Sahan Chowdhury

Comparative Analysis of Standard Backpropagation Algorithm Implementations: MATLAB vs Python for Rice Type Classification

Description and Motivation

Rice type classification appears as a crucial task within the food and agriculture industries. In 2023 alone, nearly 510 million metric tonnes of rice were produced [1] Classification task can be streamlined through creating algorithms which recognise rice types based on their parameters, this was proven successful where machine learning algorithms were used for classification tasks in the agriculture and food industry [2]. This project showcases the implementation of a standard backward propagation algorithm with a single hidden layer to compares training speeds and test set accuracies between MATLAB and Python on a rice type dataset.

Initial Analysis of Dataset

The dataset compromises of 18,185 entries and were classed as: '0' - Gonen and '1'- Jasmine. Jasmine rice accounted for 54.9% of the data. There are 12 attributes, of which 10 are numerical features which are useful to classify the rice type. Initial exploratory data analysis indicated that the dataset should be normalised due to its vast range amongst different variables. There was no missing values nor outliers significant enough to be removed. Box-Cox transformation was then applied to make the data follow more closely to the normal distribution and mitigate the effect of skewness. Finally, the dataset was split into test and train using a holdout method in a 70:30 split.

Comparison

Python:

To make the comparison fair both the MATLAB and Python implementations used only a sigmoid activation function. Hyper-parameters: learning rates which were examined were 0.01, 0.1 and 0.3 this was done to cover a range of values high to low. The number of neurons assessed were 10, 20 and 30. The number of epochs which was used was 10 this seemed to be the most computational efficient and has the best convergence. Upon initial training, training speeds in python were mostly consistent across the different hyper parameters as shown in figure (1). Higher training speeds were observed when number of neurons was 30, however an exception to this was learning rate 0.01, hidden = 10 which took the longest of 0.051 seconds. The average train speed in python being 0.018 seconds. Looking at the testing accuracies as shown in figure (2) the highest accuracy percentage was 55.1% observed within 3 hyperparameter combinations (hyperparameter: LR: 0.01, Hidden: 10, LR: 0.01, Hidden: 20 & LR: 0.1, Hidden: 30). Although there was no evident pattern hidden neurons = 30 seemed to have the highest accuracy consistently while also having higher average training times of 0.022 seconds. Learning rate 0.3 had the lowest accuracy overall.

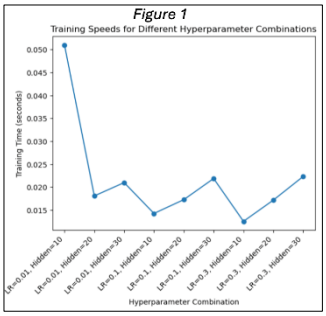

Figure 1: A graph showing the training speeds for different hyperparameter combinations.

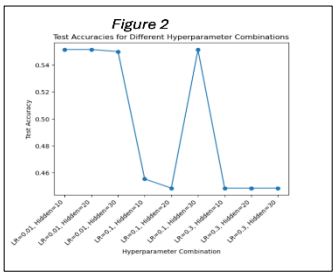

Figure 2: A graph showing the test accuracies for different hyperparameter combinations.

MATLAB:

The average test accuracy across all hyperparameter combinations obtained was 0.283, which is nearly half from the one obtained using Python. From the results obtained, it was observed that when the learning rate was 0.3, there was a substantial increase in the test accuracy, which could suggest that the network converges faster. Moreover, the training time remains constant at 0.27s. Furthermore, it was observed that the hyperparameter combination which produced the highest test accuracy was the one where the learning rate was 0.30 and the hidden layer size was 20. Moreover, the confusion matrix was calculated for the network, as well the ROC curve. From the ROC curve it was calculated that the AUC was 0.51, which indicates that the network could narrowly predict the rice type better than random chance.

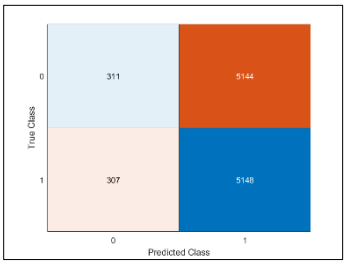

Figure 3: A confusion matrix for the MATLAB implementation.

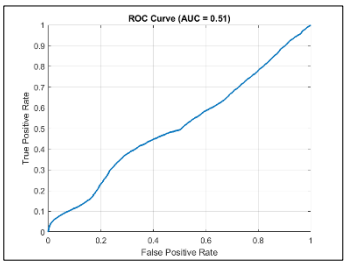

Figure 4: A graph showing the ROC curve.

Critical Evaluations

As seen in the previous sections of the report, it could be concluded that Python performed better than MATLAB in this task. This could be down to a few reasons, for example in MATLAB, the network might have overfitted to the training sets. Another reason could be that Python libraries are often at the forefront of implementing the latest machine learning algorithms and optimizations.

Lessons Learned & Future Work

In this study, we learned the importance of selecting the optimal hyperparameters for the networks, in order to achieve the best results and accuracies. In future work, we should seek to do a more in-depth data pre-processing, experiment with other algorithms and implement cross-validation.

References:
[1] T. text provides general information S. assumes no liability for the information given being complete or correct D. to varying update cycles and S. C. D. M. up-to-Date D. T. R. in the Text, "Topic: Rice," Statista. https://www.statista.com/topics/1443/rice/#topicOverview
[2] Mustafa Tasci, Ayhan Istanbullu, Selahattin Kosunalp, T. Iliev, I. Stoyanov, and I. Beloev, "An Efficient Classification of Rice Variety with Quantized Neural Networks," Electronics, vol. 12, no. 10, pp. 2285–2285, May 2023, doi: https://doi.org/10.3390/electronics12102285.
[3] https://www.mdpi.com/2079-9292/12/10/2285 [4] https://www.sciencedirect.com/science/article/pii/S0168169921003021?casa_token=6Jz73JidQY8AAAAA:-9iAlOjewAPFSUeWfv19s57B1iYVhsE4KAPlrws1C0TjZaBo9Zwf5mn7MpAGliKUS6TCNDp1aA