# Comparing Standard Backpropagation Algorithm in Matlab and Python

Syed Mahbubul Huq (220033725) | Ibtida Sadiq (210049741)
City, University of London

## *Introduction*

For the task of comparing standard backpropagation algorithms in Matlab and Python, we decided to do a regression task and predict the demand for rented bikes for a particular region by considering different climate and environmental factors.

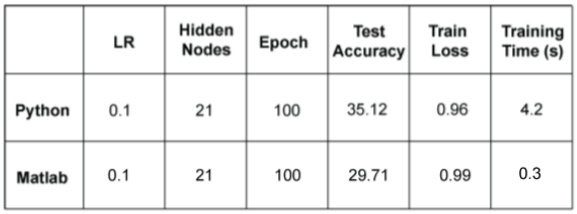
## *Dataset and Related Works*

'Seoul Bike Sharing Demand Data Set' from UCI Machine Learning Repository [1] was selected. The dataset contains 14 attributes and 8760 instances in total. By forming a heatmap, we were able to find out highly correlated columns for our task. We found the 'Dew point temperature(°C)' and 'Temperature(°C)' columns have a correlation of 0.91, so we omitted one column.

Among papers using similar datasets [2], the authors used SVM, GBM, and XGBT algorithms, among them GBM showed the highest accuracy. In another paper [3], the author used a different dataset with similar attributes to predict bike-sharing demand. They used a Deep LSTM algorithm having two layers of LSTM in their mode. Their model performed better than other proposed models like DNN, LSTM+DNN, etc.

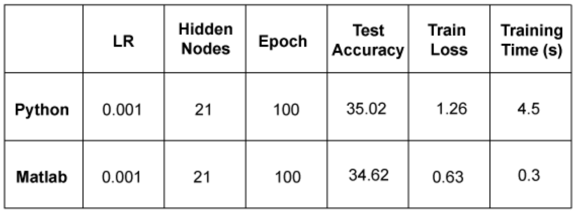
## *Comparison and Critical Evaluation*

For different changes of factors and variables, we get different results in Matlab and Python. In the following table, a comparison is presented:



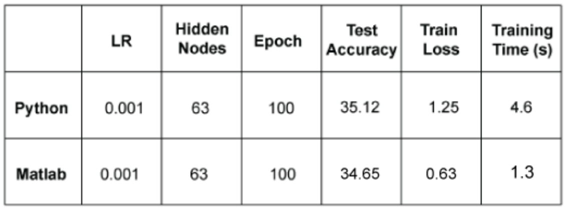
| | LR | Hidden Nodes | Epoch | Test Accuracy | Train Loss | Training Time (s) |
|---|---|---|---|---|---|---|
| Python | 0.1 | 21 | 100 | 35.12 | 0.96 | 4.2 |
| Matlab | 0.1 | 21 | 100 | 29.71 | 0.99 | 0.3 |

Changing the learning rate from 0.1 to 0.001 we get the following result:



| | LR | Hidden Nodes | Epoch | Test Accuracy | Train Loss | Training Time (s) |
|---|---|---|---|---|---|---|
| Python | 0.001 | 21 | 100 | 35.02 | 1.26 | 4.5 |
| Matlab | 0.001 | 21 | 100 | 34.62 | 0.63 | 0.3 |

Lastly, we see a comparison by changing the hidden nodes number from 21 to 63 and keeping the learning rate to 0.001:



| | LR | Hidden Nodes | Epoch | Test Accuracy | Train Loss | Training Time (s) |
|---|---|---|---|---|---|---|
| Python | 0.001 | 63 | 100 | 35.12 | 1.25 | 4.6 |
| Matlab | 0.001 | 63 | 100 | 34.65 | 0.63 | 1.3 |

The test accuracy using Python is always more compared to Matlab. With the change in learning rate and hidden nodes, accuracy and train loss can be seen. The algorithm performed best in Python having a higher learning rate and lower hidden nodes, whereas, for Matlab, the highest test accuracy was achieved with a lower learning rate and higher hidden nodes. A significant difference can be seen in the case of training time, in every case Matlab was faster than Python. This is because matrix multiplication is faster in Matlab compared to Python. In the case of backpropagation while calculating weights, matrix multiplication is carried out which is significantly higher in speed compared to Python operation. Besides, a very significant change in train loss is seen in the case of Python and Matlab, while changing the variable factors. Matlab performed better in case of train loss by reducing the learning rate and increasing the hidden nodes number To get a better understanding the following graphs are presented which deal with the data in our last table.

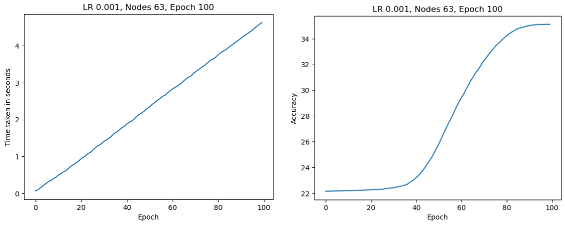


*fig1: Change of Time and Accuracy for Python (LR 0.001, Nodes 63)*

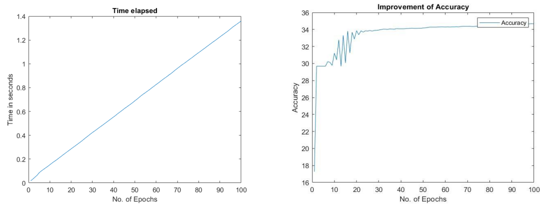


*fig2: Change of Time and Accuracy for Matlab (LR 0.001, Nodes 63)*

## *Lessons Learned and Future Work*

Varying factors like the number of nodes, learning rate, etc, and comparing results on different platforms can have significant effects on accuracy and loss value. Having this lesson we would like to apply different algorithms and change various factors like epoch number, LR, and node numbers, to see the changes to achieve higher accuracy as there is plenty of scopes since we have lower accuracy.

## *References*

[1]https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand?fbclid=IwAR1fjJU1wthZ--Q49n09k1fvJZRJ7mrond_JQIATiDTiY5Qb_vJVLfIlJgo

[2]https://www.sciencedirect.com/science/article/pii/S0140366419318997?fbclid=IwAR3ET9gQ_Dp_pcNRSfjB6nObB1XIWa8lqdtecSbxa19N0UPAPxVFZ79Ujig

[3]https://www.sciencedirect.com/science/article/pii/S1877050919302364?ref=pdf_download&fr=RR-2&rr=7a1e52c10b0874d5