Andika William
16/398497/PA/17458

MLP Implementation on Iris Dataset

1. Case Explanation

In our previous assignments, we implemented an SLP model of feedforward artificial neural network. That is, a neural network that consists of an input layer and an output layer. Though the implementation is simpler, an SLP typically fails in reaching certain accuracy standards. In this case a multiple layer of classification is necessary to gain a more accurate prediction. By principle, each layer added in a feed-forward multi-layer perceptron adds a new level of non-linearity that is not possible in a single layer.

For this assignment, we will be implementing a Multi-Layer Perceptron on the Iris dataset. Inside this dataset, there are 4 input layers that will be used (sepal/petal width and length). There will be only one additional hidden layer that will be put in between the Input layer and the Output layer, as illustrated by *Figure 1.*

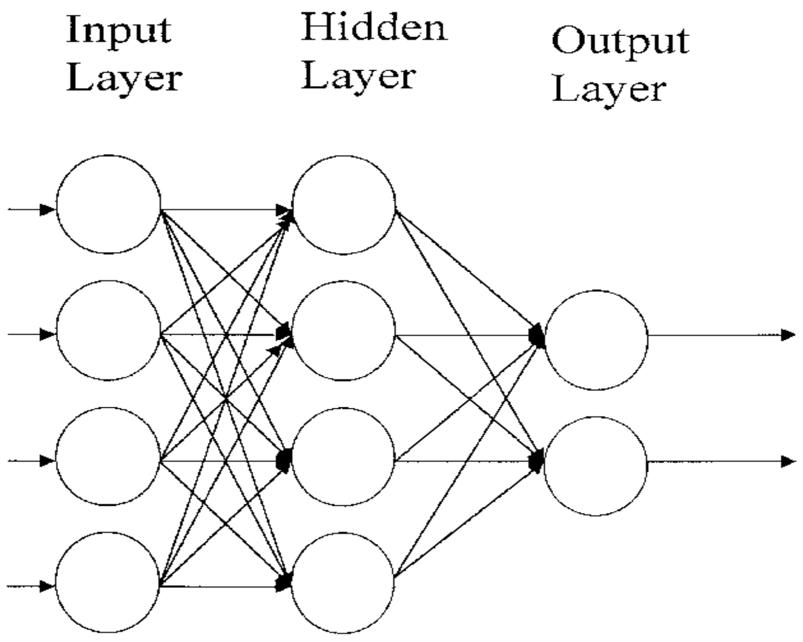


Figure 1.

2. Implementation

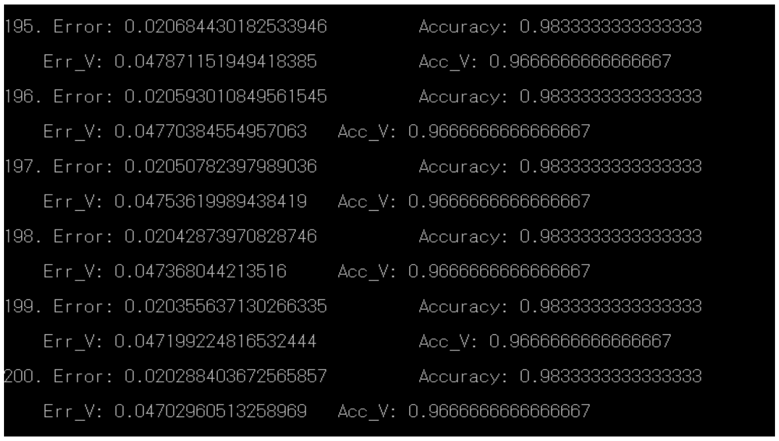
In writing the code, the language that was used in this assignment was Python (version 3.7.1). It was written in the Sublime text editor and was tested in the cmd.

https://github.com/andkwv/MPL_Iris

3. Experiment & Result

Running the code:

The model will be trained and validated by using the split method. Out of the 150 tuples in the dataset, 30 will be set aside randomly for validation. The rest, 120, will be used for training the model. This time, our model will go through 200 epochs of training.



```
195. Error: 0.020684430182533946        Accuracy: 0.9833333333333333
   Err_V: 0.047871151949418385          Acc_V: 0.9666666666666667
196. Error: 0.020593010849561545        Accuracy: 0.9833333333333333
   Err_V: 0.04770384554957063   Acc_V: 0.9666666666666667
197. Error: 0.02050782397989036         Accuracy: 0.9833333333333333
   Err_V: 0.04753619989438419   Acc_V: 0.9666666666666667
198. Error: 0.02042873970828746         Accuracy: 0.9833333333333333
   Err_V: 0.047368044213516     Acc_V: 0.9666666666666667
199. Error: 0.020355637130266335        Accuracy: 0.9833333333333333
   Err_V: 0.047199224816532444          Acc_V: 0.9666666666666667
200. Error: 0.020288403672565857        Accuracy: 0.9833333333333333
   Err_V: 0.04702960513258969   Acc_V: 0.9666666666666667
```

Figure 2.

4. Result :

As expected, this model was able to converge at much better accuracy compared to the Single-Layer implementation. The model was able to predict the category of Iris with almost perfect accuracy (96 %). This value happens in both the training data and the validation data.

However, though the accuracy is satisfactory, there are inconsistencies in performance. For example, one instance of running may see that the model converges on a 60-70% accuracy (*Figure 5.)*, while others show convergence at different rates. We can conclude that this is caused by the varying number of initialization values. Because of unrestricted random values of 0 until 1, the performance of the model will vary greatly.

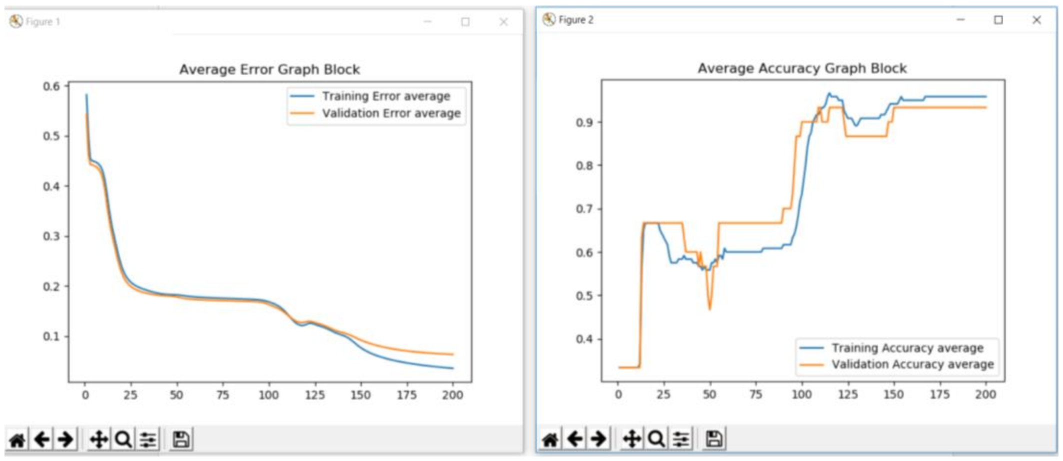


Figure 3.

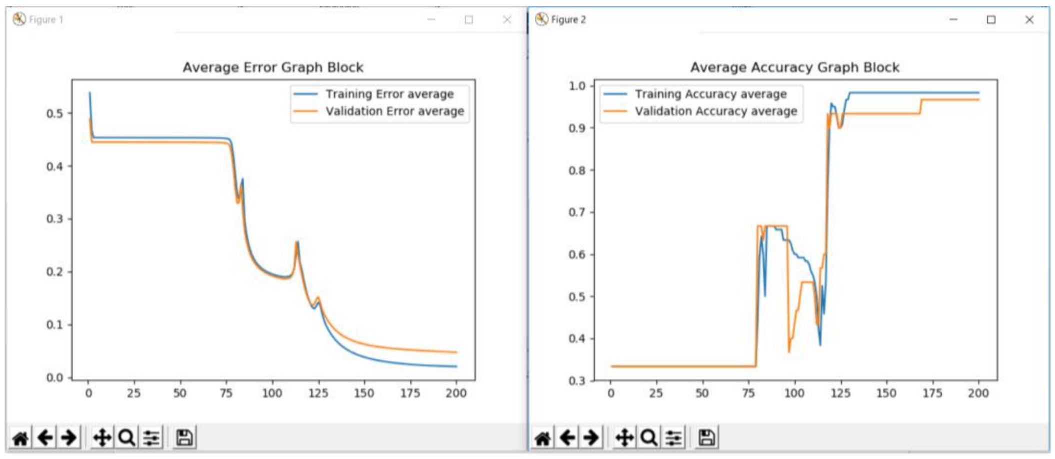


Figure 4.

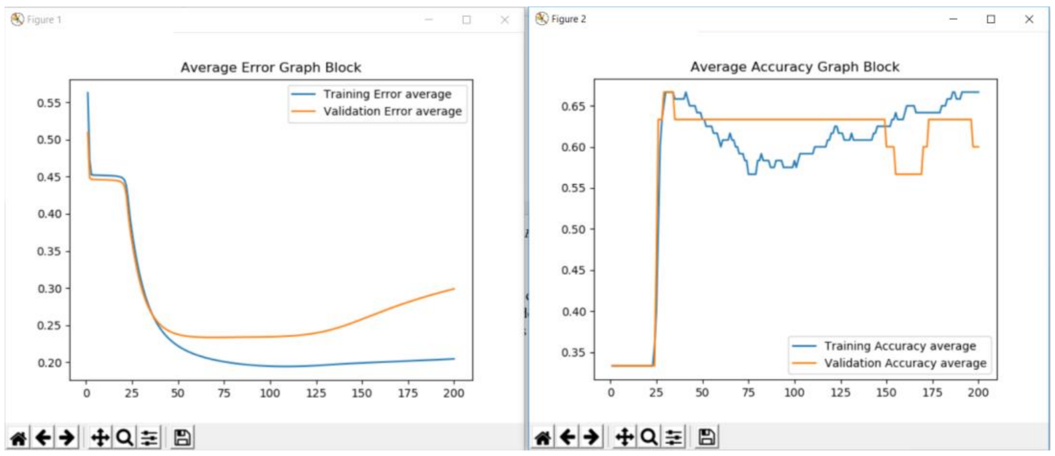


Figure 5

5. Conclusion

Through our results, we can conclude that the performance of the MLP model can predict with better accuracy compared to single layer type of model. However, the random initial values will greatly affect the performance of the model. There comes a need to restrict the initial random values for the input.