# **Dry Beans Classification using Multi-Layer Perceptron**

## **Task Description**

The dataset in this task contains 5 features (Area, Perimeter, MajorAxisLength, MinorAxisLength, Roundness) and 3 classes (BOMBAY, CALI, SIRA). The aim of this task is to build a fully connected multi-layer perceptron neural network that can distinguish among the three different classes using all 5 features.

## **Preprocessing**

* This dataset does not require much preprocessing given the nature of the features provided.
* The features were normalized using a standard scaler to avoid overflow issues when multiplying big numbers.
* Null values were filled by the mean value of their respective column.

## **Analysis**

### **#1 Using Hyperbolic Tangent Activation Function**

## **A screenshot of a computer Description automatically generated**

**(BEST RESULTS)**

Using one hidden layer or even two layers with about 5 nodes per layer in addition to using bias with a learning rate of 0.01 and at least 100 epochs yields 100% most of the time, although it may vary because of the random weights assignment.

A screenshot of a computer

Description automatically generated

Using unsuitable parameters like a low number of epochs and maybe even no bias does not allow the network to converge fully, it even fails to classify 8 entries out of 60.

A screenshot of a computer

Description automatically generated

Using too much hidden layers and neurons makes the network perform much worse, most likely because of vanishing gradients problem.

### **#2 Using Sigmoid Activation Function**

A screenshot of a computer

Description automatically generated

**(BEST RESULTS)**

Sigmoid function seems to be performing similar to the hyperbolic tangent function, it achieves a perfect score with an accuracy of 100%, although it needs a much higher learning rate (0.1 vs 0.01) and two hidden layers with 3 and 4 nodes respectively.

A screenshot of a computer

Description automatically generated

Using more than one hidden layer achieves lower accuracy, also turning off bias results in an even lower accuracy of 33%

A screenshot of a computer

Description automatically generated

Using a very simple neural network that only has one hidden layer and one node fails to differentiate between the three classes, it can tell the difference between BOMBAY and other types of beans, but it can’t differentiate among SIRA and CALI beans as seen from the confusion matrix.

## **Best Accuracy for each Activation Function**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Activation Function | Train Accuracy | Test Accuracy | Learning Rate | #Epochs | #Layers | #Nodes |
| Sigmoid | 100% | 100% | 0.1 | 100 | 2 | 3 4 |
| Tanh | 100% | 100% | 0.01 | 100 | 1 | 5 |

## **Conclusion**

* Hyperbolic tangent performs slightly better than sigmoid function when it comes to differentiating between the three bean classes.
* Using very deep neural networks with many layers and neurons performs worse than using less on this simple task.
* Insufficient number of epochs or too low learning rate significantly hinders the convergence of the model.
* Using bias greatly affects the outcome, with using bias tending to achieve better results
* Very simple and shallow networks are not complex enough to yield a successful multiclass classification outcome.