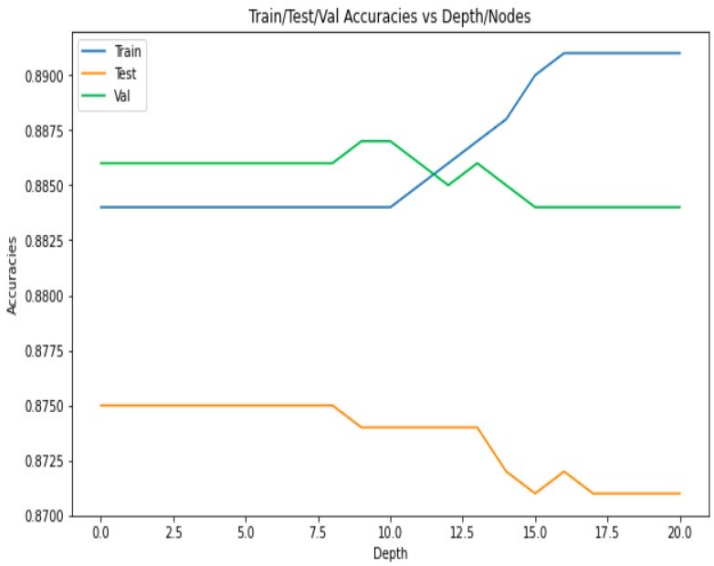
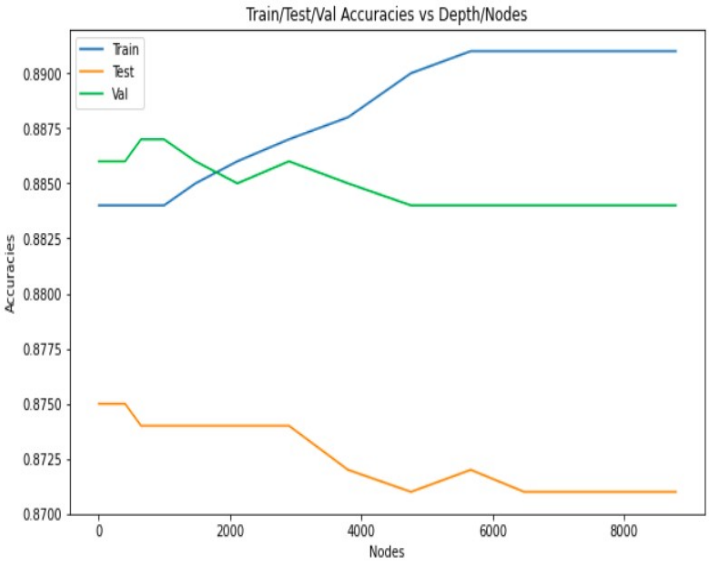
COL774 – Machine Learning (Assignment-3)

Prakhar Aggarwal
2019CS50441

Part 1 (Decision Trees and Random Forests)

(a) (i) Accuracies analysis with depth and nodes (multi-way split)





For depth 20:

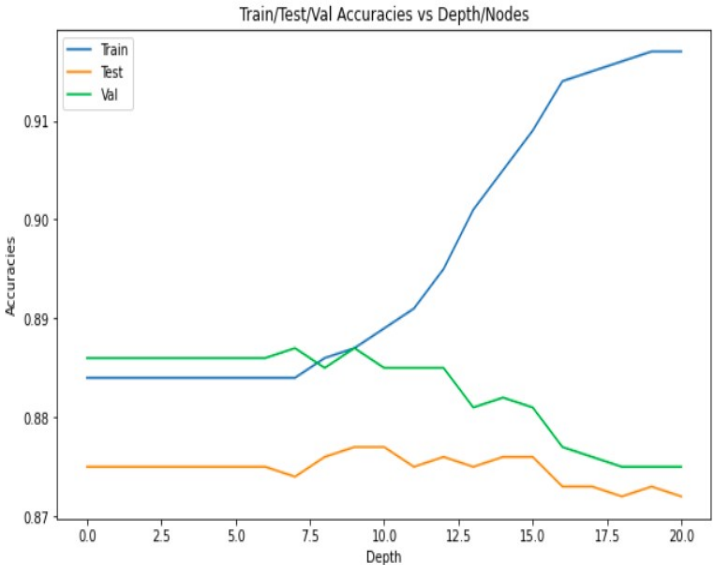
Training accuracy: 89.1%
Testing accuracy: 87.1%
Validation accuracy: 88.4%

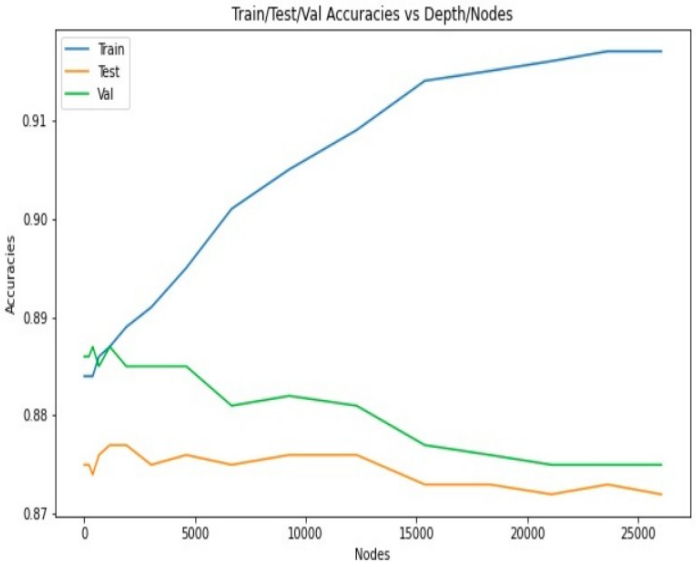
As can be seen from the curve, the model tends to overfit for higher depth since training accuracy is continuously increasing whereas the testing as well as the validation accuracy is decresing.

Max validation accuracy is observed for depth = 9, after this point, test as well as validation accuracy tends to decrease.

Accuracies at depth = 9
Training accuracy: 88.4%
Testing accuracy: 87.4%
Validation accuracy: 88.7%

(a) (ii) Accuracies analysis with depth and nodes (one hot encoding)





<u>For depth 20:</u>

Training accuracy: 91.7%
Testing accuracy: 87.2%
Validation accuracy: 87.5%

As can be seen from the curve, the model tends to overfit for higher depth since training accuracy is continuously increasing whereas the testing as well as the validation accuracy is decresing.

Max validation accuracy for one hot encoding is also observed for depth = 9, after this point, test as well as validation accuracy tends to decrease.

Accuracies at depth = 9
Training accuracy: 88.7%
Testing accuracy: 87.7%
Validation accuracy: 88.7%

(b) Post pruning

For this part, depth was set to 9 (because of good results in previous part). Using bredth first search, all nodes as well as the node count was computed. Out of all nodes, best node (node which results in max increase in validation accuracy) was picked and pruned.

It takes _ iterations to maximize the validation accuracy. Number of nodes reduce from _, _, _, _, _, _.

(c) Random Forests

Part 2 (Neural Networks)

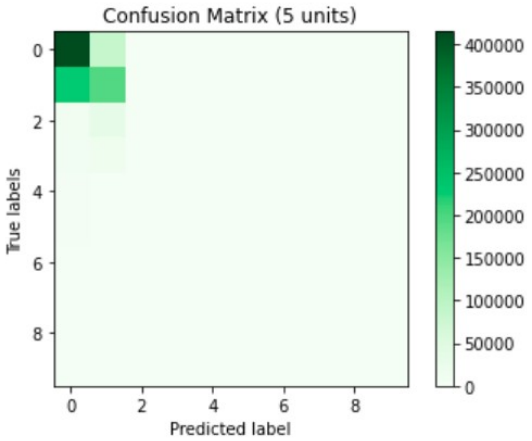
(a) In this part, categorical features are transformed to their one hot encoding representation. The new dataset obtained contains 85 features.

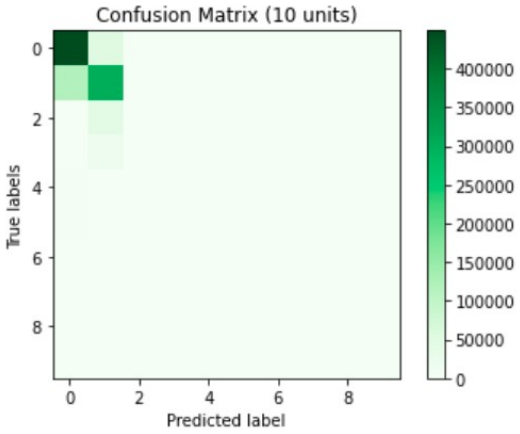
(b) In this part, a neural network model was constructed from scratch. Forward propagation, backward propagation were coded from first principles. Mini-batch SGD was used to faster the learning process.
The model is general enough to take the following values as parameters:
Mini-batch size, Number of features, Hidden layer architecture, Number of target classes, learning rate, epsilon, Number of Epochs.

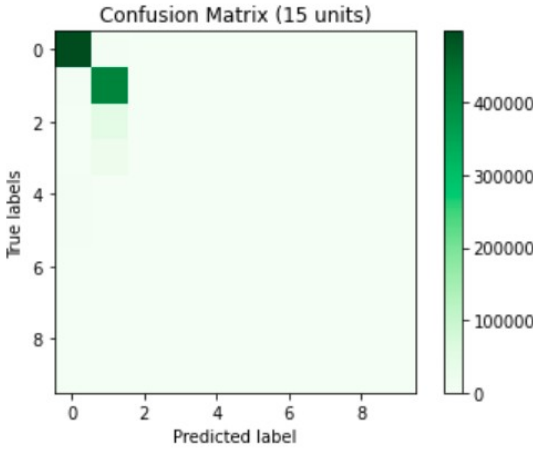
(c) Convergence Criteria:

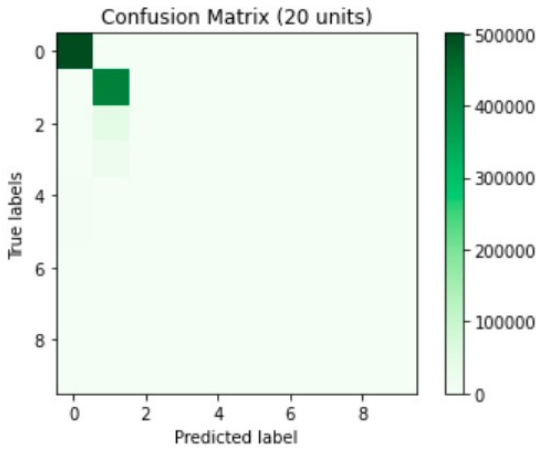
- There is a hard bound on the number of the epochs (1000)
- There is a variable epsilon which looks at the difference of errors of consecutive epochs and stops if this difference is less than epsilon (0.0001)
- We have a variable MIN_LOSS which helps in avoiding convergence in a case where error difference of consecutive epochs is very small but error of a particular epoch is high.
- Learning rate: 0.1

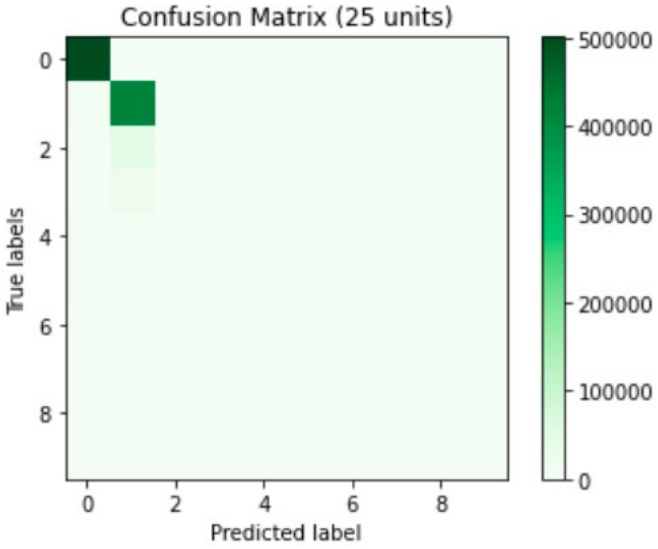
Num of Units	5	10	15	20	25
Training Time	87.866s	81.766s	88.185s	91.792s	96.754s
Train Accuracy	62.687%	76.997%	91.975%	92.275%	92.225%
Test Accuracy	61.022%	75.169%	91.503%	92.225%	91.984%

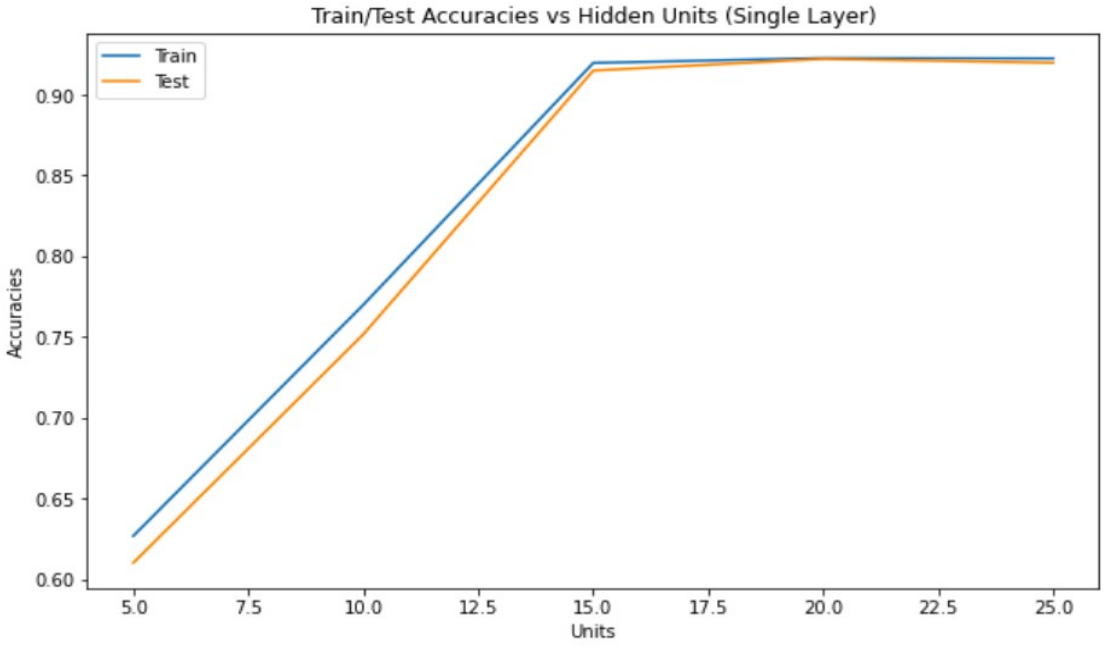





Confusion Matrix (15 units)
True labels
Predicted label
0
2
4
6
8
400000
300000
200000
100000


Confusion Matrix (20 units)
True labels
Predicted label
0
2
4
6
8
500000
400000
300000
200000
100000


Confusion Matrix (25 units)
True labels
Predicted label
0
2
4
6
8
500000
400000
300000
200000
100000


Train/Test Accuracies vs Hidden Units (Single Layer)
Train
Test
Accuracies
Units
0.60
0.65
0.70
0.75
0.80
0.85
0.90
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0