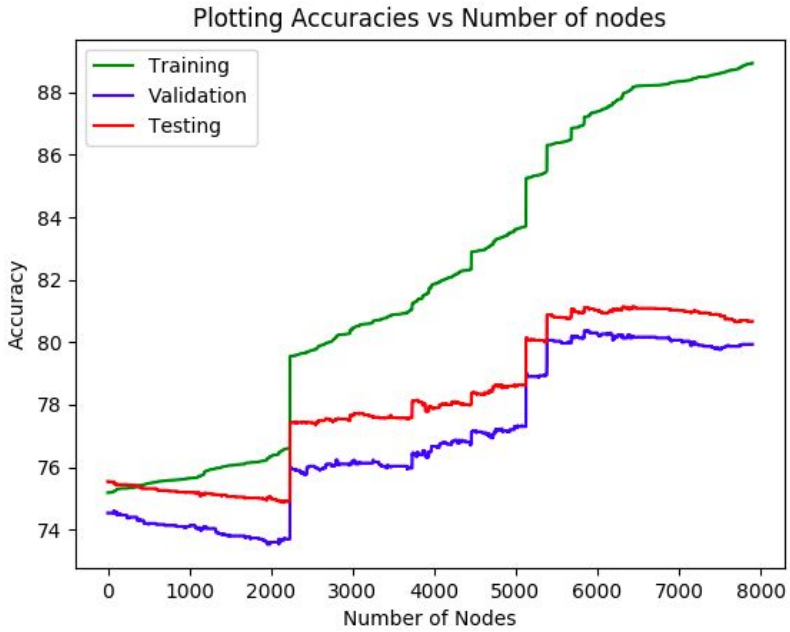
COL774: Machine Learning

Name: Mahima Manik
Entry Number: 2017MCS2093

Q. 1

a) Total Nodes formed - 7902
Maximum Height - 11
Training accuracy - 88.93333333333334
Validation accuracy - 79.93333333333334
Testing accuracy - 80.67142857142858



Initially, as we increase the number of nodes, all training, validation and testing accuracies increases. At some threshold, training accuracy continues to grow but validation and testing accuracies saturate. After the threshold, if we further increase the number of nodes, tree overfits on training data and accuracy starts on decreasing for validation as well as testing data.

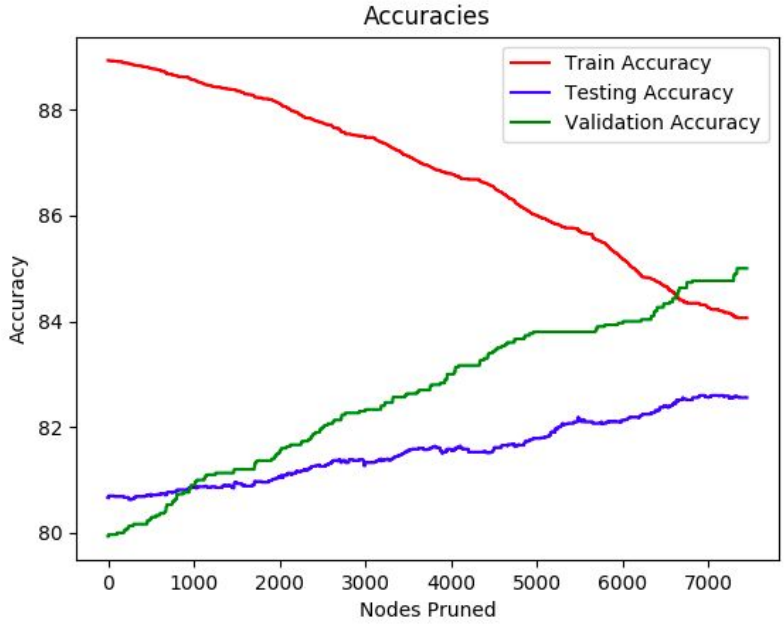
b) **Pruning**

We prune the nodes in the tree in bottom-up fashion. The nodes on whose removal, accuracy on validation data set remains same or does not increase is pruned. If it's parent becomes leaf node, then it is also considered for pruning in further iterations.

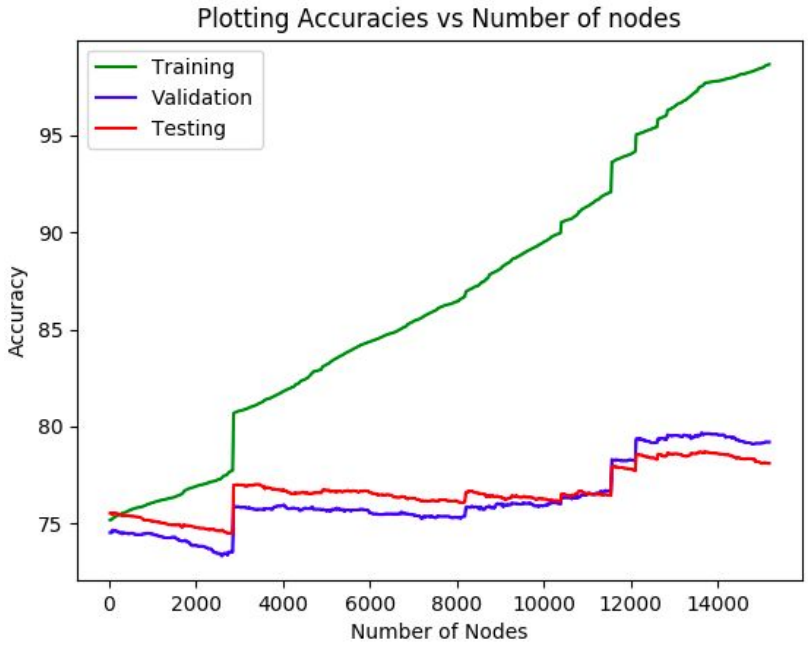
Total Nodes after pruning - 454
Training accuracy - 84.06296296296296
Validation accuracy increases from 79.93333333333334 to 85.0

Testing accuracy - 82.55714285714286

We observe that significant numbers of nodes are getting pruned. Considering validation data accuracy for pruning also improves performance on training data.



c) Number of nodes: 15196
Training accuracy: 98.66666666666667
Validation accuracy: 79.2
Testing accuracy: 78.1



This model cleary overfits on training data as number of nodes is almost double from part (a) and training accuracy is also very high. Validation and Testing accuracy is also lesser that part (a) due to overfitting at training. It overfits because we are allowing the continuous features to repeat on one path/branch, which increases the number of nodes drastically.

Printing 6 branches from the root gives the following results:

Relationship Education Age 32.0 Occupation Hour per Week 40.0 Race Work Class Sex Age None	Relationship Education Occupation Native Country Work Class Race Hour per Week 40 Age 54.0 Age None	Relationship Education Occupation Native Country Age 34.5 Age None	Relationship Marital Status Education Occupation Work Class Fnlwgt 181896 Fnlwgt 119028.5 Fnlwgt 157935.0 Age None	Relationship Education Occupation Age None	Relationship Occupation Education Age None

d) Parameters**:** `max_depth, min_samples_split, min_samples_leaf`

Parameters (11, 2, 3) -
Training accuracy: 86.77037037037037
Validation accuracy: 85.1
Testing accuracy: 84.5

Parameters (12, 3, 2) -
Accuracies - 87.21481481481482, 85.3, 84.28571428571429

Parameters (12, 8, 3) -
when chosen from range ((1, 20), (2, 10), (1, 10))
Accuracies - 87.05925925925926, 85.39999999999999, 84.5

Validation accuracy just increases by 0.4% from part b), so doesn't show much difference but training and testing accuracies increases by 3-4%. This because in part b) we have optimized our tree for the validation data, while pruning. So, here training and testing accuracy increases whereas validation shows no much difference.

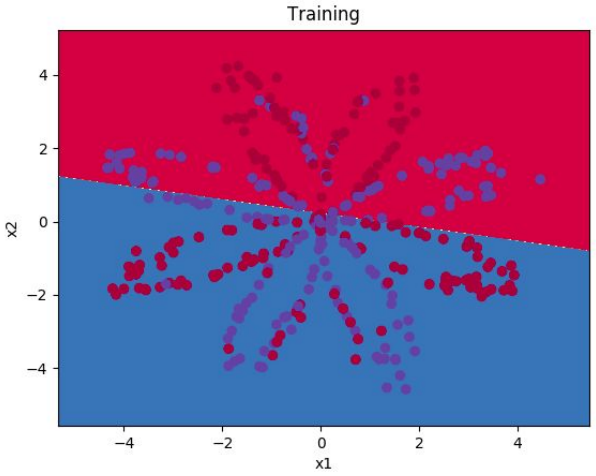
e)
Training accuracy: 86.63333333333333
Validation accuracy: 85.43333333333332
Testing accuracy: 85.85714285714286
Parameters chosen for best accuracy are:

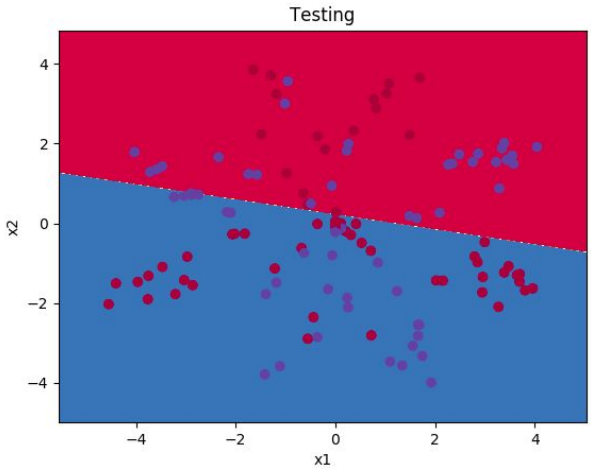
- Maximum depth for each tree = 10
- Bootstrap = True
- N_estimators = 5
- max_ features = 14
- min_samples_split = 3
- min_samples_leaf = 4

On making the Bootstrap=False, the accuracy decreases to (86.24814814814815, 84.96666666666667, 84.81428571428572).
On increasing the maximum depth for each tree, the accuracy increases by 3% (89.2074074074074, 85.6, 85.5142857142857)
Increasing the number of trees in the forest, does not give any noticeable change in the accuracy.
Decreasing the number of features in max_features decreases the accuracies

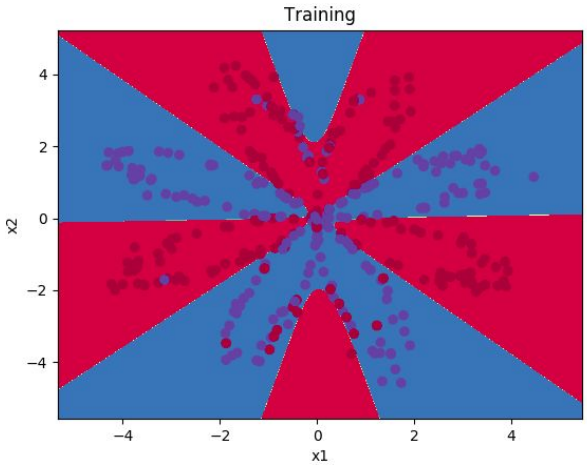
It is giving better accuracies compared to part (b). Training accuracy is better in part c), but it is overfitting on training and giving lesser accuracy on validation and testing data as compared to this.

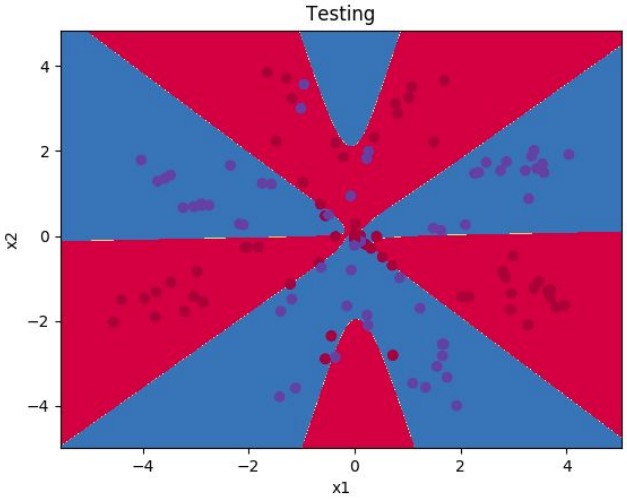
Q. 2
a) This part is implemented by creating class object for each layer in the model and storing deltas, weights and activations as arrays.
b)
i) Outputs from sklearn *linear_model.LogisticRegression* is as follows:
Training accuracy : 45.78947368421053
Testing accuracy : 38.333333333333336





ii) Batch Size: 380 = number of examples
List of numbers of perceptron: [5]
Training Accuracy: 90.0
Testing Accuracy: 84.16666666666667





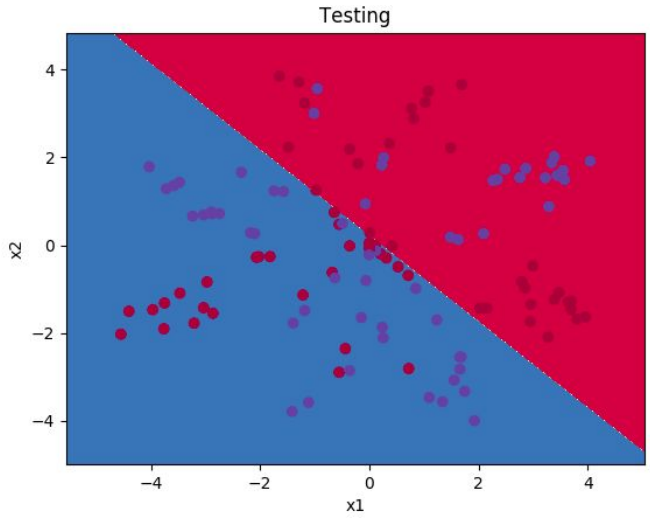
iii)

As we increase the number of hidden units, the accuracy on training data increases. Decision boundary also becomes more clear on points, but testing accuracy decreases on too much increasing the hidden layers. For instance, on 3 hidden perceptrons, the testing accuracy was 85% but on 10 hidden perceptrons, testing accuracy drops to 82%.

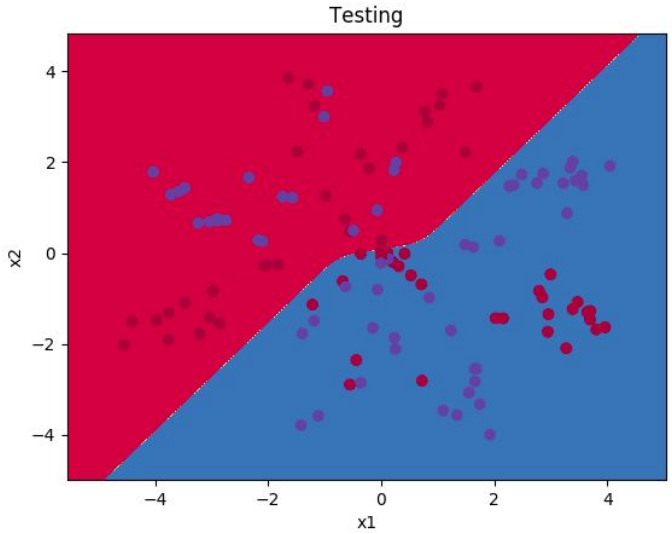
We get best training as well as testing accuracy on 20 perceptrons on single hidden layer.

Hidden Layer: [1]
Training Accuracy: 63.68421052631579

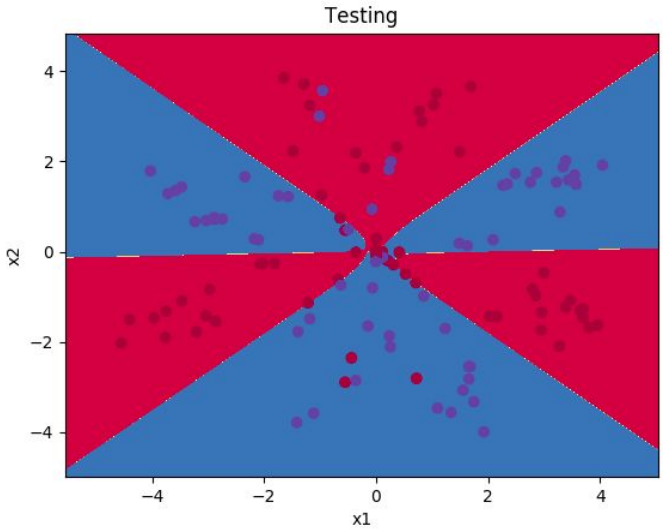
Testing Accuracy: 56.666666666666664



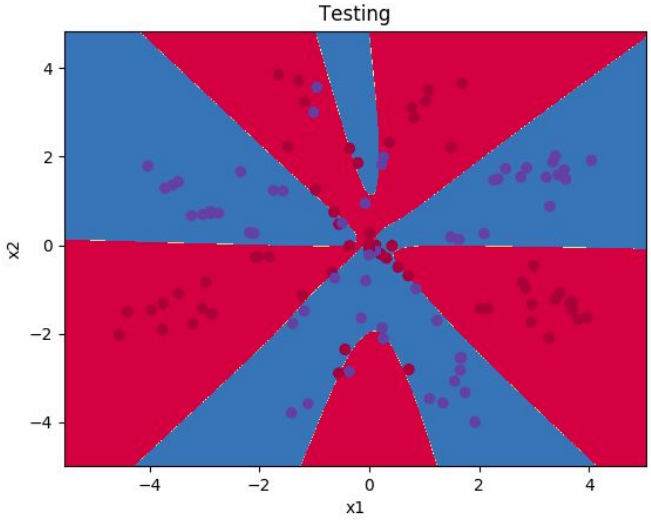
Hidden Layer: [2]
Training Accuracy: 60.26315789473684
Testing Accuracy: 56.666666666666664



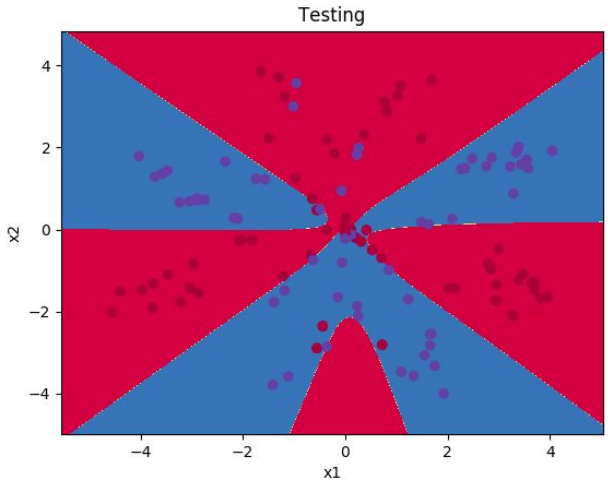
Hidden Layer: [3]
Training Accuracy: 89.21052631578948
Testing Accuracy: 85.0



Hidden Layer: [10]
Training Accuracy: 91.57894736842105
Testing Accuracy: 82.5



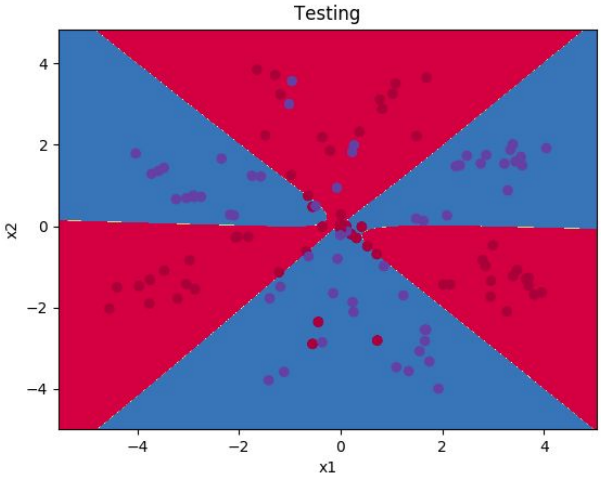
Hidden Layer: [20]
Training Accuracy: 89.21052631578948
Testing Accuracy: 86.66666666666667



Hidden Layer: [40]

Training Accuracy: 89.73684210526316

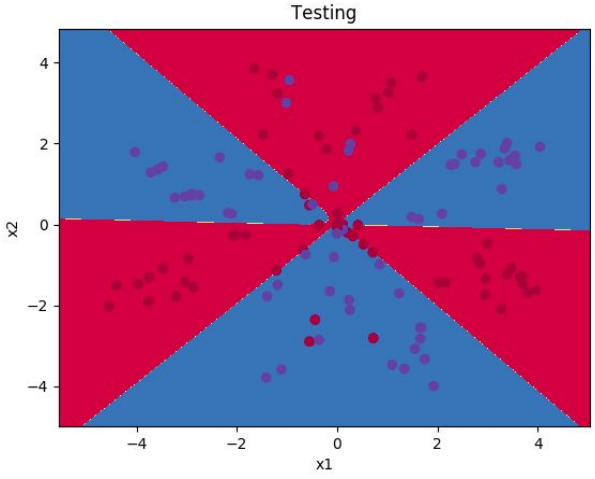
Testing Accuracy: 85.83333333333333



iv) Hidden Layer: [5, 5]

Training Accuracy: 90.0

Testing Accuracy: 85.83333333333333



c)

i) In LIBSVM, we used multi-class classification for binary. Kernel type is taken as linear.
C = 1
Testing Accuracy = 99.87%
Testing Accuracy = 98.8056%
Even with linear kernel, the accuracy is pretty high. It means that the result is linearly separable.
Accuracies from our neural implementation:
Training Accuracy: 70.06
Testing Accuracy: 69.41666666666667

ii) 100 perceptrons in Hidden Layer

Training Accuracy: 97.48
Testing Accuracy: 97.47222222222223
Stopping criteria is taken as when the absolute value of difference between error of present bach and previous batch is less than 0.015.

```
Convergence Criteria: (abs(batch_err-prev_err) > 0.015)
```

iii) RELU

Training Accuracy: 97.57000000000001
Testing Accuracy: 97.63888888888889
Improvement is very less as compared to using Sigmoid function. Accuracy was pretty high even with Sigmoid.