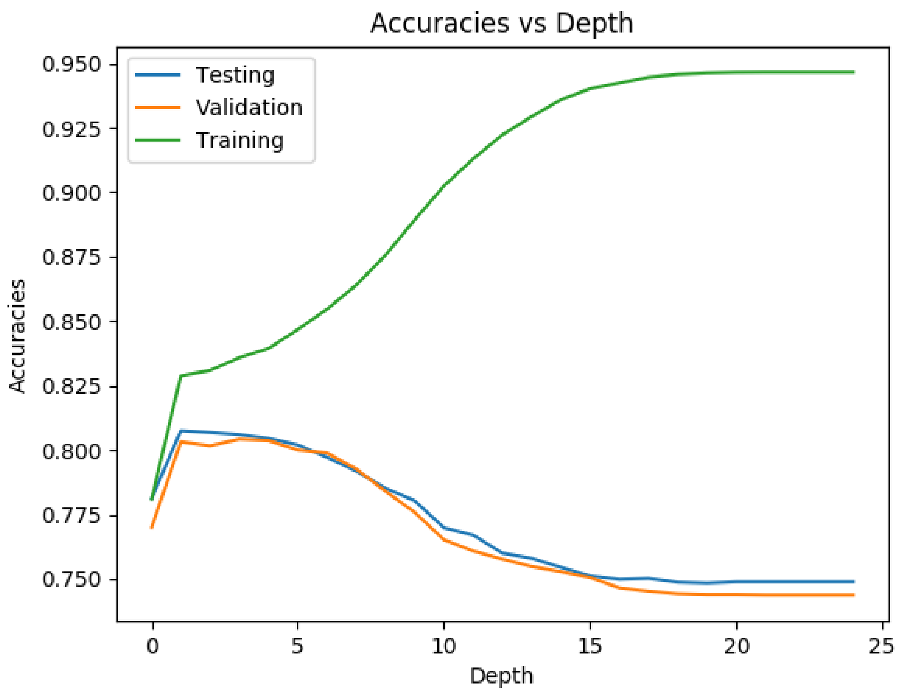
Assignment-3 Report

- Pradyumna Meena (2016CS10375)

Decision Tree

Q 1



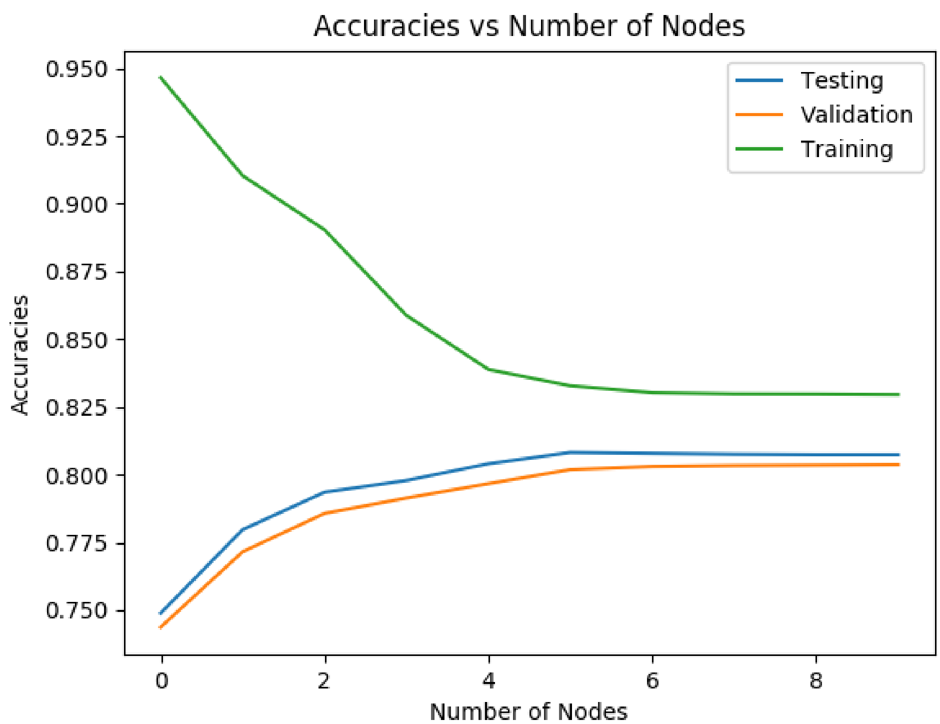
Results:

1. Training Set Accuracy -> 0.9465
2. Test Set Accuracy -> 0.7488
3. Validation Set Accuracy -> 0.7433

We can see that the model tends to overfit after a certain point of time since accuracy on train and test set keep on increasing and decreasing respectively. That point occurs at depth of 3, after which the accuracy starts to decrease. At this depth accuracies for train,test and validation set respectively are 0.8358, 0.806 and 0.8041.

Q 2

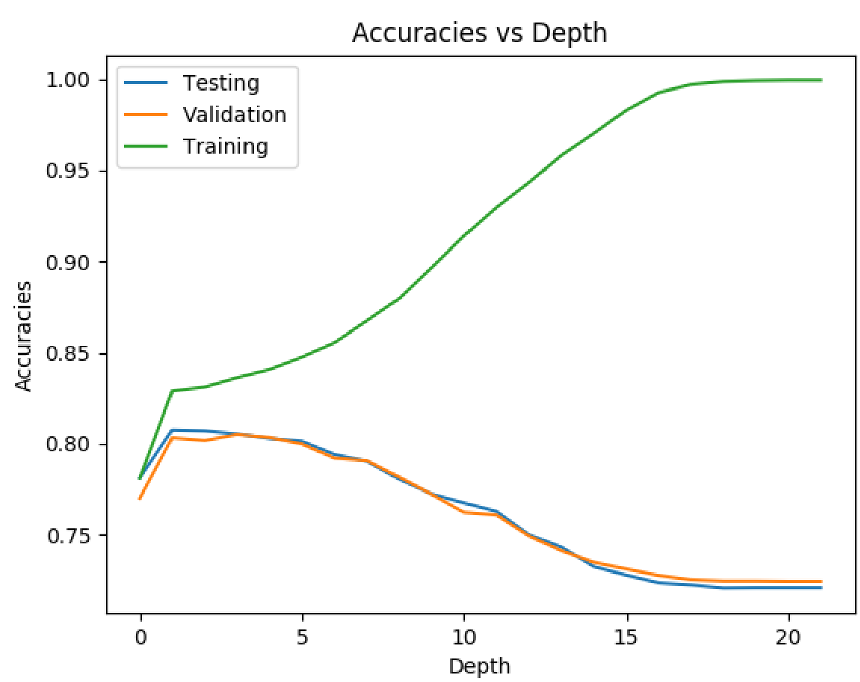
It takes 9 traversals through all set of nodes to reach the break point. The number of nodes reduce from 8659 to 5256 to 3621 to 2096 to 929 to 195 to 55 to 44 to 42 to 40. On X-axis I have used the iteration number or the epoch. Initially the accuracies change by a significant amount but then it saturates to a steady value.



Results:

1. Training Set Accuracy -> 0.8295
2. Test Set Accuracy -> 0.8073
3. Validation Set Accuracy -> 0.8036

Q 3

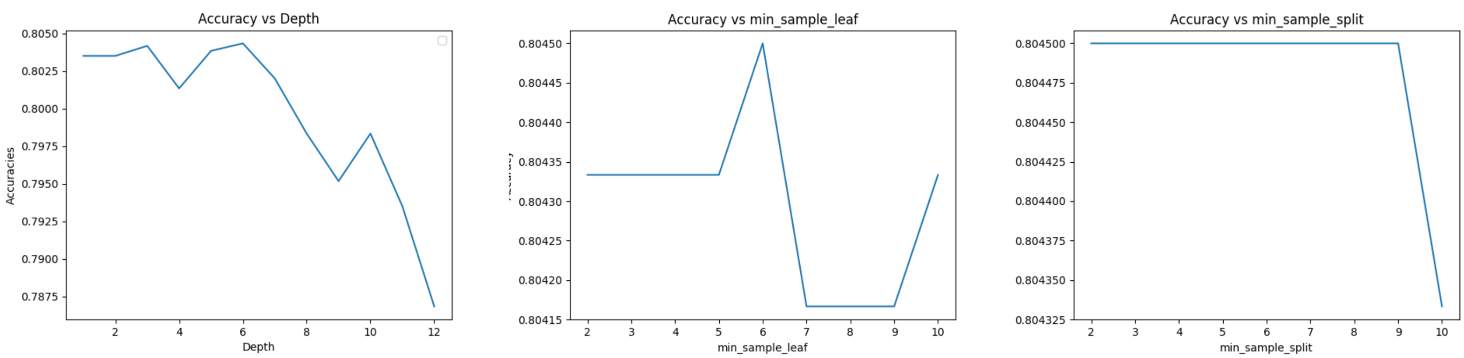


Results:

1. Training Set Accuracy -> 0.9992
2. Test Set Accuracy -> 0.721
3. Validation Set Accuracy - > 0.7245

This clearly shows that we are overfitting and even more than in Q 1. Hence we have much higher accuracy on training data while reduced accuracies on test and validation sets. This is because at every step we are trying to fit the data in more better way as possible by modifying the data based on the incoming variation and not on the basis of some initial calculation.

Q 4



Parameters associated with best accuracy are as followed:

1. depth = 6
2. min_sample_leaf = 6
3. min_sample_split = 3

Results:

1. Training Set Accuracy -> 0.8372
2. Test Set Accuracy -> 0.8088
3. Validation Set Accuracy -> 0.8045

This set of parameters gives more of uniform accuracies. Neither does it overfit nor does it underfits the given data. The results are very much like those observed in Q2. As we can see variation in min_sample_split and min_sample_leaf is not having very significant effect on accuracy as compared to the depth.

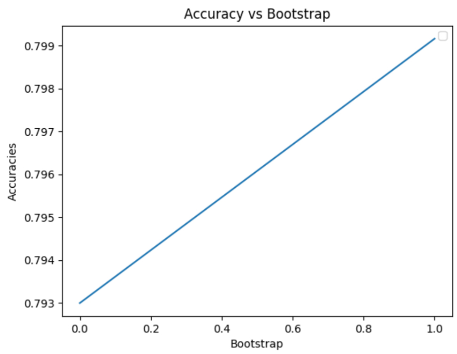
Q 5

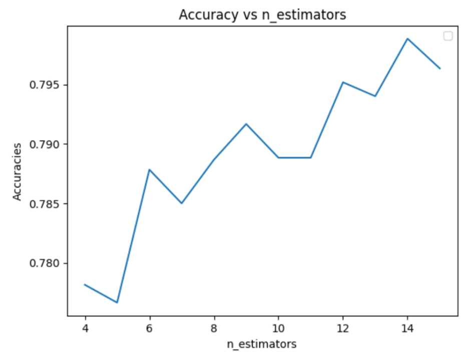
Results:

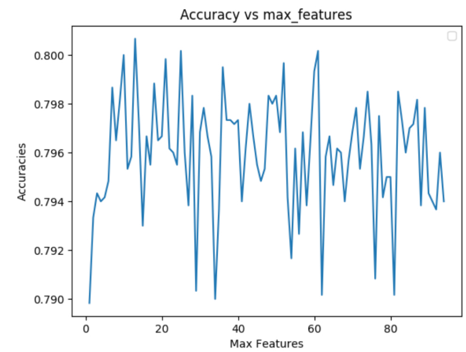
1. Training Set Accuracy -> 0.8351
2. Test Set Accuracy -> 0.8048
3. Validation Set Accuracy -> 0.7986

The results are very much like as observed in the previous part as well as Q2. However they are very much different than Q3.

Q 6







Parameters associated with best accuracy are as followed:

1. num_estimators = 14
2. bootstrap = True
3. max_features = 52

Results:

1. Training Set Accuracy -> 0.9996
2. Test Set Accuracy -> 0.7966
3. Validation Set Accuracy -> 0.7905

The model ends up overfitting on training data like in Q3 though performs slightly better than it on test and validation data.

Neural Nets

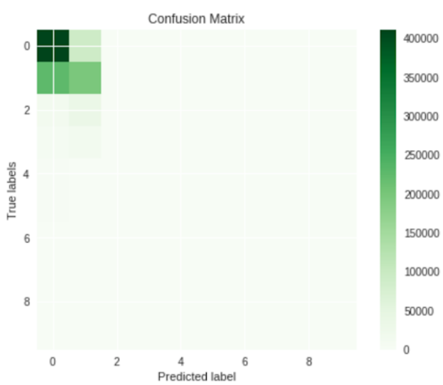
Convergence criteria:

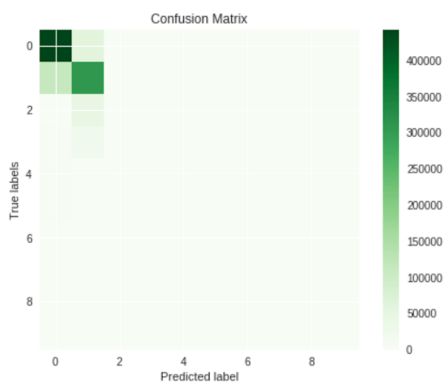
- Num_epochs > max_allowed
  - max_allowed = 1000 for sigmoid
  - max_allowed = 3000 for relu
- Training_loss < 10^{-6}
- Learning_rate < 10^{-10} (in case of adaptive learning)

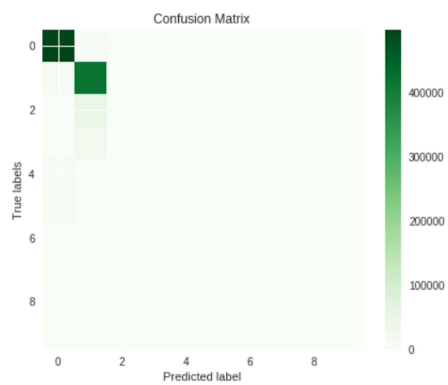
Sigmoid Activation (1 hidden layer):

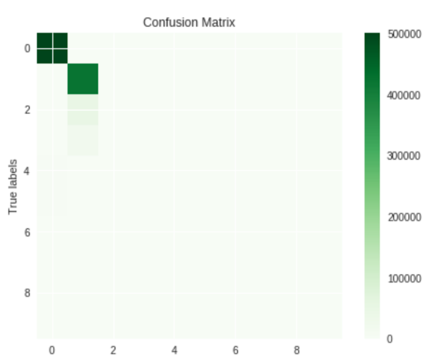
Neurons	5	10	15	20	25
Training Time	133.9	151.4	162.34	172.33	191.25
Training Accuracy	0.6229	0.7722	0.9188	0.9230	0.9228
Test Accuracy	0.6071	0.7510	0.9144	0.9221	0.9205

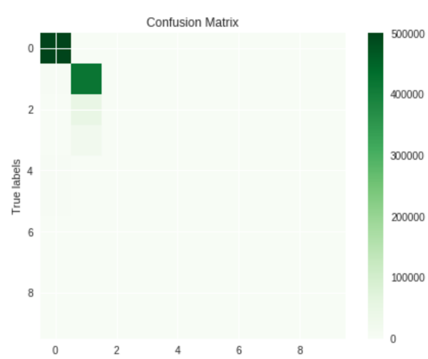
Below are confusion matrices for testing data for single layer with 5,10,15,20,25 neurons respectively







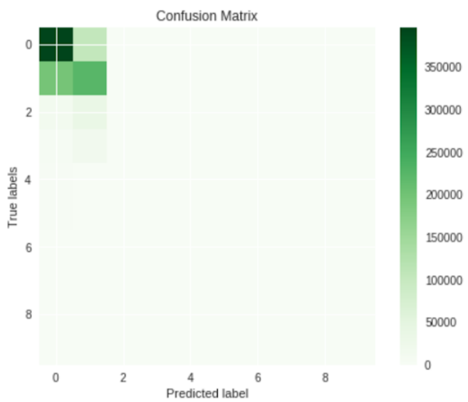


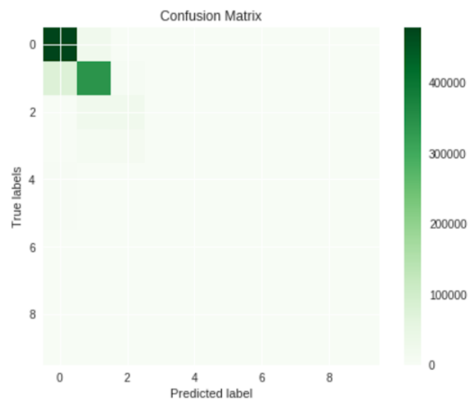


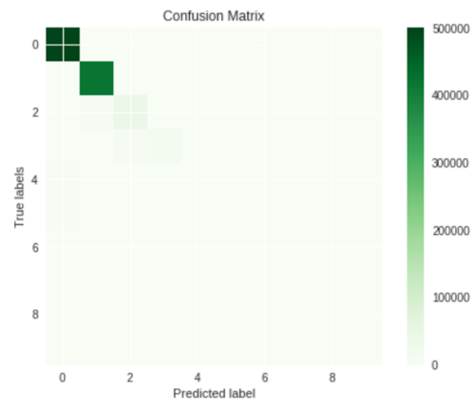
Sigmoid Activation (2 hidden layers):

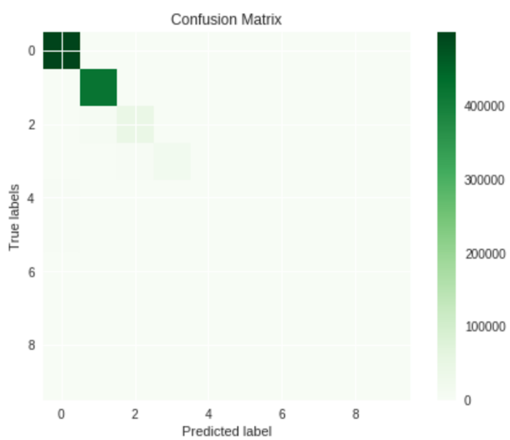
Neurons	5	10	15	20	25
Training Time	170.11	196.83	227.55	249.79	272.96
Training Accuracy	0.6377	0.8600	0.9856	0.9918	0.9916
Test Accuracy	0.6223	0.8432	0.9772	0.9820	0.9769

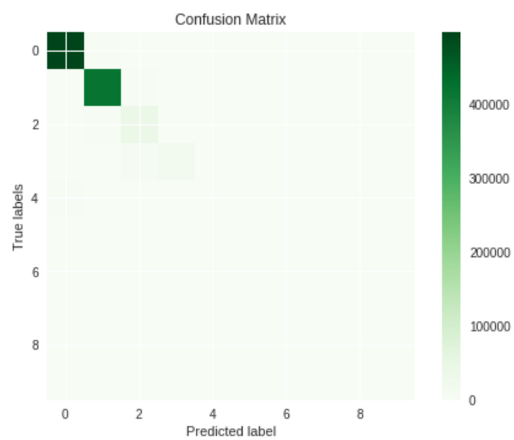
Below are confusion matrices for testing data with architecture: {5,5},{10,10},{15,15},{20,20},{25,25} neurons



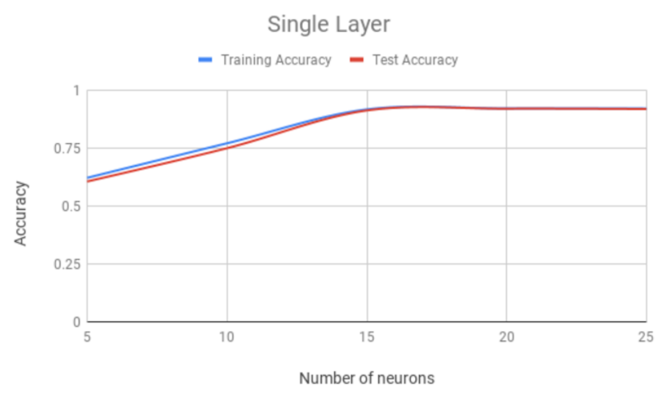


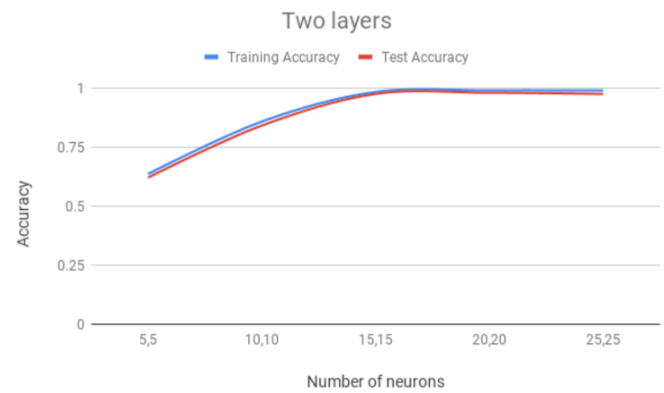




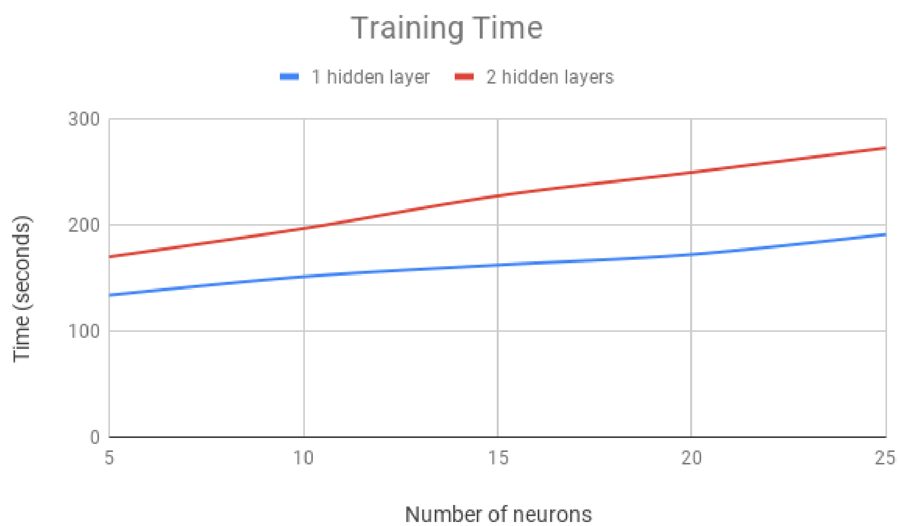


Comparison for Sigmoid activation





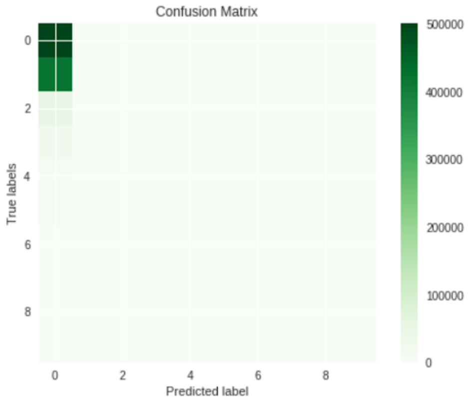
Above shows the variation of accuracy as we increase number of neurons in layers and below is the training times of the various architectures

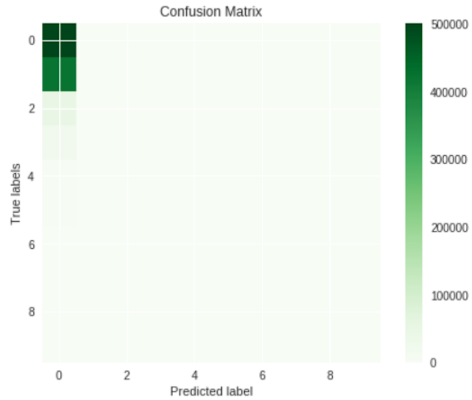


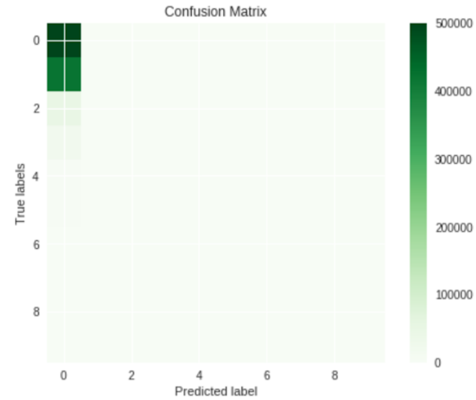
Adaptive Learning (1 hidden layer + Sigmoid Activation):

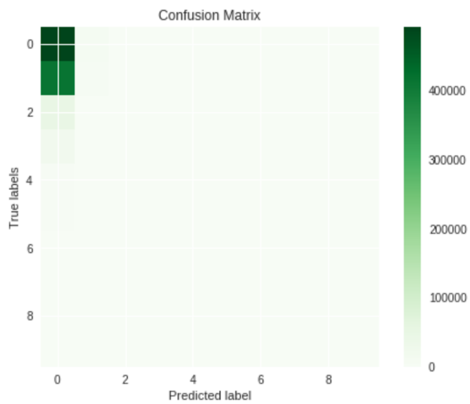
Neurons	5	10	15	20	25
Training Time	1.97	2.18	2.4	2.56	2.72
Training Accuracy	0.4995	0.4994	0.4994	0.4986	0.4993
Test Accuracy	0.5012	0.5011	0.5010	0.4994	0.5007

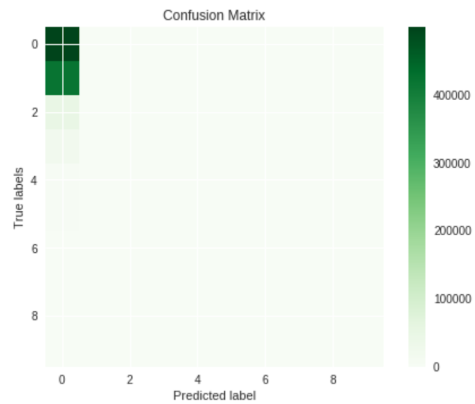
Below are confusion matrices for testing data for single layered architecture with 5,10,15,20,25 neurons





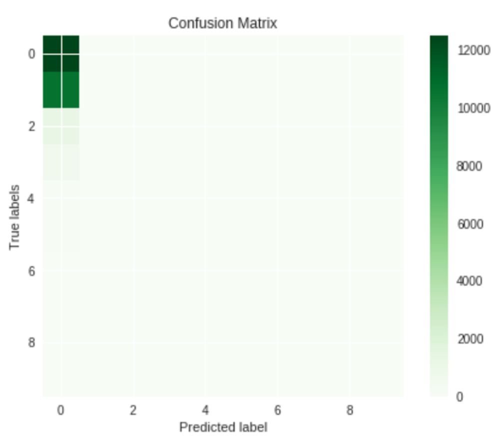


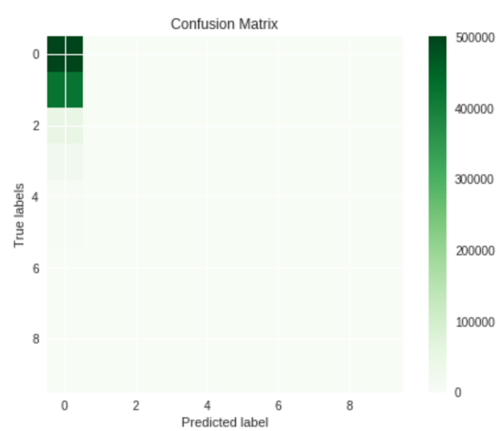




Adaptive Learning (2 hidden layers + Sigmoid Activation):

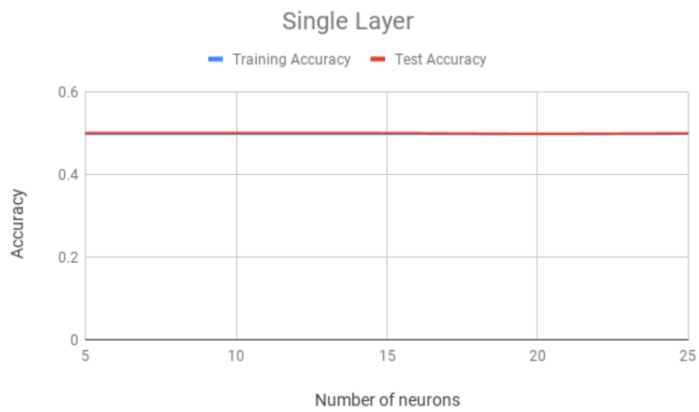
Neurons	5	10	15	20	25
Training Time	2.53	2.92	3.34	3.64	3.95
Training Accuracy	0.4995	0.4995	0.4995	0.4995	0.4995
Test Accuracy	0.5012	0.5012	0.5012	0.5012	0.5012

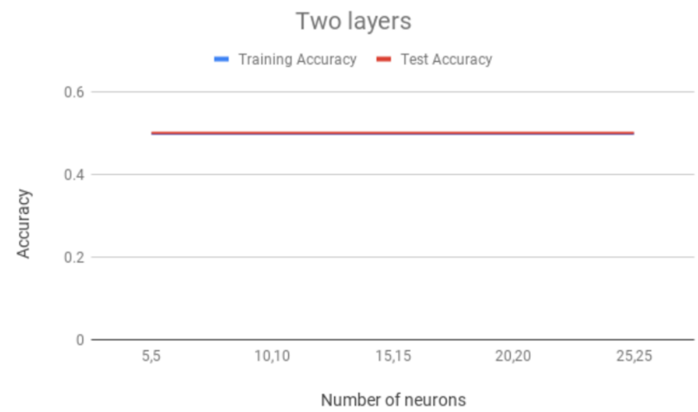




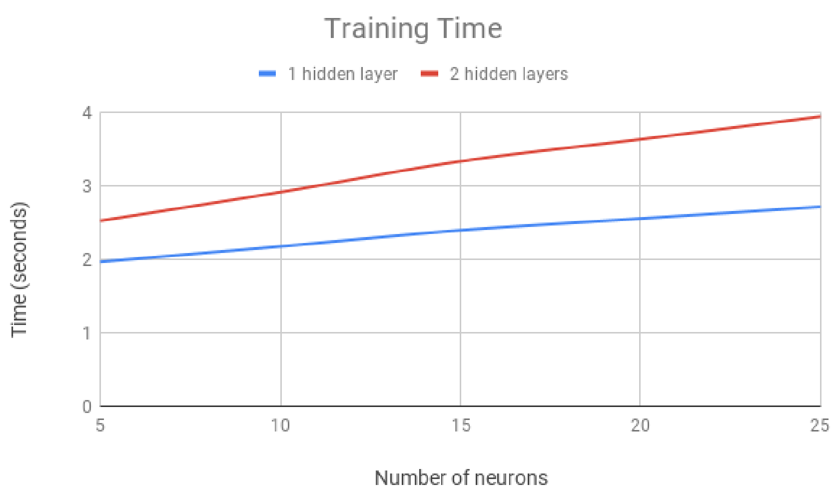
On the left is confusion matrix for training data and on left is for testing data. All architectures showed exactly same results.

Comparison for Sigmoid activation with adaptive learning





Above shows the variation of accuracy as we increase number of neurons in layers and below is the training times of the various architectures

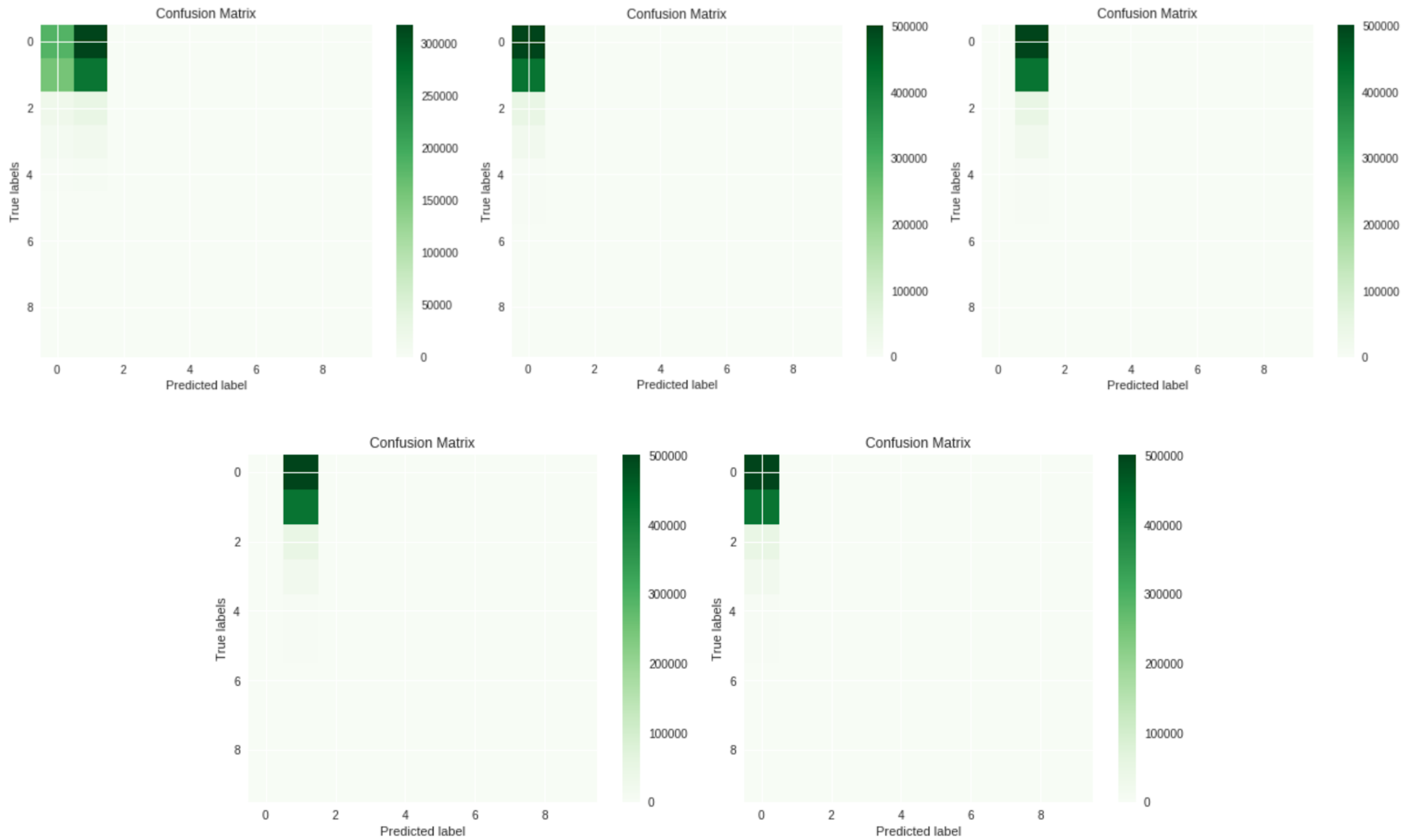


In ReLU I faced the problems of dying neurons i.e. the neurons which are not getting activated because some huge gradient updates passed through it and it is never able to give a positive output.

Adaptive Learning (1 hidden layer + Adaptive Learning + ReLU Activation):

Neurons	5	10	15	20	25
Training Time	1.765	1.847	1.933	1.962	2.064
Training Accuracy	0.4518	0.4995	0.4238	0.4237	0.4995
Test Accuracy	0.4495	0.5012	0.4224	0.4224	0.5012

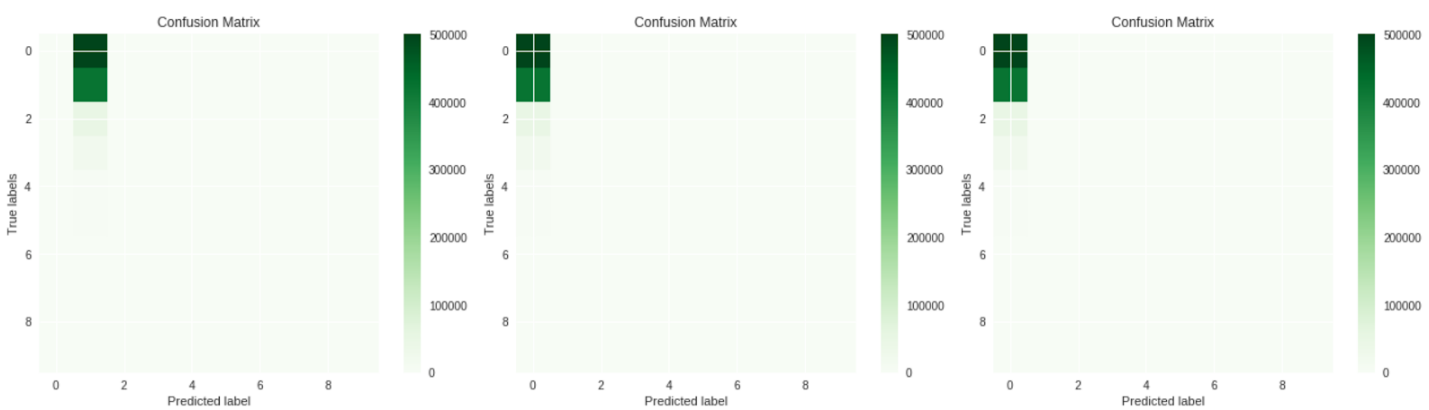
Below are confusion matrices for testing data for 5,10,15,20,25 neurons respectively.

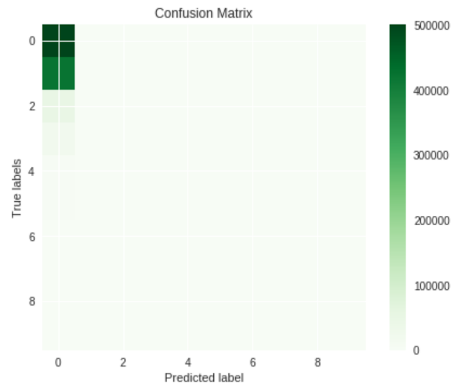


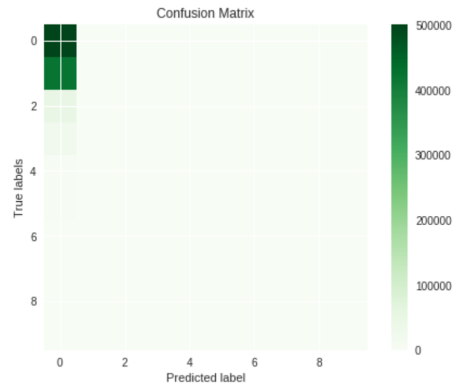
Adaptive Learning (2 hidden layers + Adaptive Learning + ReLU Activation):

Neurons	5	10	15	20	25
Training Time	2.07	2.23	2.42	2.48	2.72
Training Accuracy	0.4237	0.4995	0.4995	0.4995	0.4995
Test Accuracy	0.4224	0.5012	0.5012	0.5012	0.5012

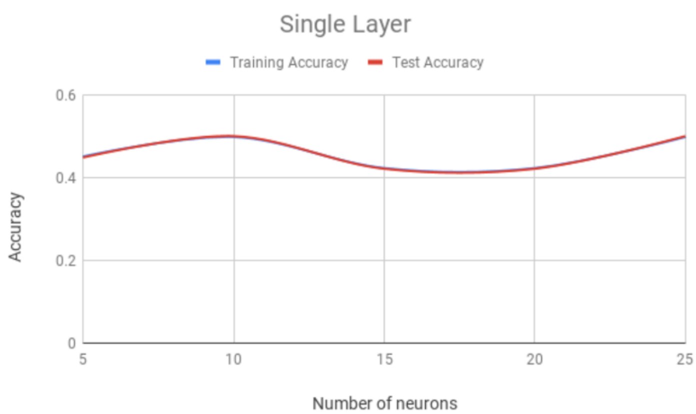
Below are confusion matrices are for testing data for testing data for 5,10,15,20,25 neurons respectively.

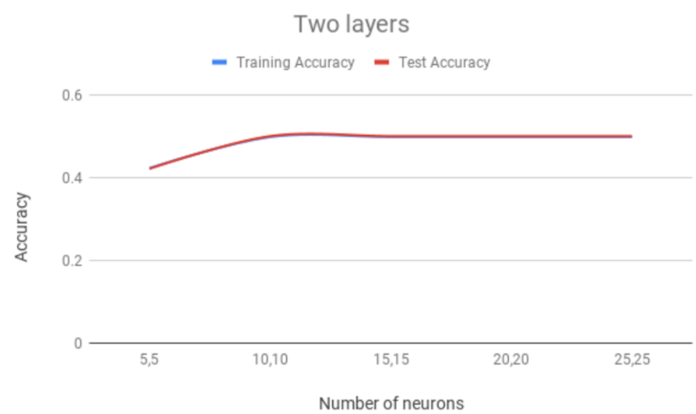




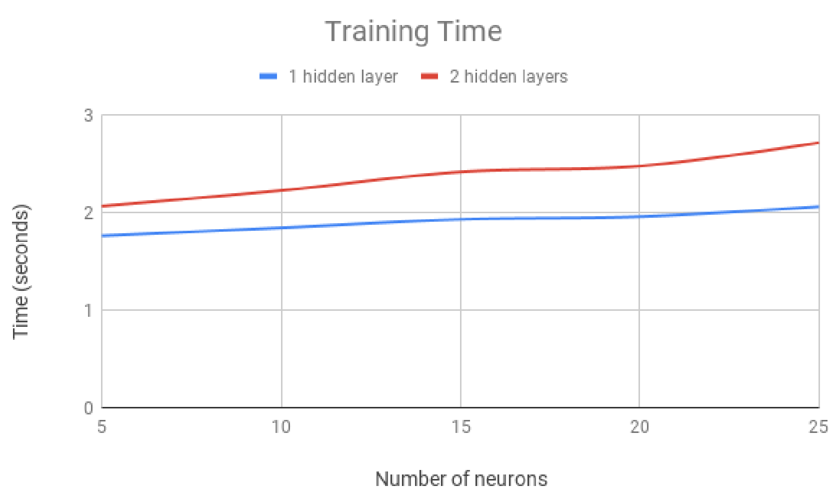


Comparison for ReLU activation with adaptive learning





Above shows the variation of accuracy as we increase number of neurons in layers and below is the training times of the various architectures



Comments

In sigmoid activation in both single and double layer the accuracy increased as we increase the number of neurons in the hidden layers. Accuracy increase upto a certain point and then saturate at around 98% or 99%. However in case of ReLU activation various dead neurons are observed and at some points gradients also explode. However as the batch_size is decreased the accuracy starts increasing for ReLU activation. In adaptive learning once the rate starts decreasing the next change in loss is also less than the minimum requirement and therefore it keeps on decreasing. Therefore a bound on learning rate was also chosen as a parameter in criteria for convergence. However apart from plain sigmoid all variations end up doing similar predictions i.e. predicting only one class and therefore accuracies are almost same for train as well as test.

Links

- One hot encoded file -> Drive Link
- Statistical Data -> Google Sheet
- Colab Notebook -> Drive Link